1982-12

Analysis of laser turbulence utilizing a video tape recorder and digital storage oscilloscope.

Connor, John Henry

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/20057

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

ANALYSIS OF LASER TURBULENCE UTILIZING
A VIDEO TAPE RECORDER AND DIGITAL
STORAGE OSCILLOSCOPE

by

John Henry Connor, Jr.

December 1982

Thesis Advisor: E. A. Milne

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)

Analysis of Laser Turbulence Utilizing a Video Tape Recorder and Digital Storage Oscilloscope

5. TYPE OF REPORT & PERIOD COVERED

Master's Thesis
December 1982

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

John H. Connor, Jr.

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

12. REPORT DATE

December 1982

13. NUMBER OF PAGES

100

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)

UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Laser, Laser Turbulence, TV, Computer Program

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The ability to measure and predict atmospheric turbulence affecting laser beam propagation is a major concern when considering military applications. Such a method using a telescope, high resolution television camera, video tape recorder, digital storage oscilloscope, and calculator system has been devised, tested and utilized. A laser beam signal is recorded on video tape for further processing. This signal is

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

displayed, stored and digitized using a Tektronix 468 digital storage oscilloscope. The digitized signal is sent to a Hewlett-Packard 9825 computing system for Fourier transform analysis and determination of the refractive index structure constant, C_n^2. Several trials were conducted using He-Ne and Ga-As lasers. The results demonstrated good correlation with theoretical predictions as well as previously analyzed data.

Approved for Public Release, Distribution Unlimited.

Analysis of Laser Turbulence Utilizing a Video Tape Recorder and Digital Storage Oscilloscope

by

John H. Connor, Jr.
Lieutenant Commander, United States Navy
B.S., North Carolina State University, 1973

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN PHYSICS

from the

NAVAL POSTGRADUATE SCHOOL
December 1982

ABSTRACT

The ability to measure and predict atmospheric turbulence affecting laser beam propagation is a major concern when considering military applications. Such a method using a telescope, high resolution television camera, video tape recorder, digital storage oscilloscope, and calculator system has been devised, tested and utilized. A laser beam signal is recorded on video tape for further processing. This signal is displayed, stored and digitized using a Tektronix 468 digital storage oscilloscope. The digitized signal is sent to a Hewlett-Packard 9825 computing system for Fourier transform analysis and determination of the refractive index structure constant, C_n^2. Several trials were conducted using He-Ne and Ga-As lasers. The results demonstrated good correlation with theoretical predictions as well as previously analyzed data.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENT

The author wishes to thank the personnel of Fort Huachuca, U.S. Army Electronics Proving Ground for their support in this project. Additionally, I want to thank Professor Ed Milne for his guidance and understanding throughout this project. His love of his work was indeed an inspiration. I would also like to thank Professor E.C. Crittenden for his aid with the laser operation and data collection. Thanks to Professor Raymond Kelly for his aid in completing the thesis. And last, but far from least, a sincere thank you to my wonderful wife, Phyl, and daughters, Kristen and Kimberly. Without their love, support, and understanding the completion of this work would not have been worthwhile.

I. INTRODUCTION

A. BACKGROUND

The increasing use of lasers and laser technology for military applications has brought about a need for analysis of the laser beam in its environment, the turbulent atmosphere. A project at Naval Postgraduate School dealing with this subject is continuing and is the main topic of this thesis.

The patterns produced by lasers on targets have inherent problems that include broadening, beam wander and intensity fluctuations brought about by turbulence in the atmosphere. These effects of atmospheric turbulence on laser propagation have been well determined [Ref. 1]. In terms of the Fried model, [Ref. 2], C_n^2, the refractive index structure constant, has been determined to adequately express the Modulation Transfer Function (MTF), or the Mutual Coherence Function (MCF) for the atmosphere.

A system has been developed that provides a measurement of C_n^2 for atmospheric turbulence along the optical path through which a laser is propagated. The system employs a

vidicon and telescope as the detector and a distant laser as the source. This system duplicates the slit scanning system presently in use at Naval Postgraduate School [Ref. 2]. Tests using the vidicon equipment have been previously completed with values for C_n^2 on the order of 10^{-15} $m^{-2/3}$ being obtained [Ref. 3]. Measurements have been made using the same experimental set-up, except that the Tektronix 468 Digital Storage Oscilloscope are used for digitization instead of the Quantex DS-30 Digital Video Processor. These measurements have demonstrated values of C_n^2 of comparable accuracy.

B. PROBLEM

By using the digital storage capability of the Tektronix 468, the previously recorded laser signal is used as an input and is evaluated by modifying the program developed by Crager [Ref. 4]. A brief overview of the procedure is described below. Detailed explantions of the experiment and computer program are contained in Chapters III and IV, respectively.

The approach taken is basically the same as that of Crager, but because of the unavailability of a disk ROM it is necessary to store both raw and processed data on

magnetic tape. The data are loaded and recorded as necessary by the HP 9825. The basic assumption permitting analysis is that the horizontal TV scan line through the laser spot is considered to accurately mirror a point spread function of the image. Final analysis using the 468 oscilloscope has shown this to be a valid hypothesis. The digitized data from the 468 oscilloscope agree with the previously measured data which use many pixels of television data digitized by the DS-30 to express the point spread function.

The sequence of analysis is that a video tape recording is made of the TV image of the propagated laser beam. The output of the video recorder is sent to the Tektronix 468, where the derived TV scan line is digitized, averaged, and stored. The HP 9825 records the digitized data and produces a line spread function (LSF), by integrating the point spread function,

$$LSF(x) = \int_{y(min)}^{y(max)} PSF(r)\, dy \qquad (1.1)$$

where

$r = \sqrt{(x^2+y^2)}$

and computes the Fourier transform of the LSF. The diffraction limited Fourier transform of the optics is now computed if the MTF of the optics has not been previously measured experimentally. Next, the program finds the MTF of the atmosphere by dividing the Fourier transform of the LSF by the Fourier transform of the optics. Finally, by curve fitting, the program computes a single value for C_n^2.

The program now predicts the size of a laser spot on a target using the calculated value of C_n^2. This value of C_n^2 is used to calculate an MTF of the atmosphere which is then multiplied by the Fourier transform of a source, and the Fourier transform of the optics. The program then calculates the inverse Fourier transform of the products of the above and uses the Abel transform to give the angular point spread intensity distribution. From this data, the fraction of energy as a function of the total energy within a given radius R is calculated.

II. THEORETICAL CONSIDERATIONS

The theory of laser beam propagation through a turbulent medium has been explained by Crittenden, and others, and is re-emphasized here for continuity purposes [Ref. 2]. Since the effects of turbulence on laser weapons is of major concern, measurement and prediction capability for these effects on laser beams is entirely relevant.

Due to the existence of reciprocity, these ideas apply to either laser designators or imagers [Ref. 5]. In the study of the theoretical model by Fried, the effects of atmospheric turbulence are investigated [Ref. 6]. This model uses the idea of a long term optical transfer function (OTF) when considering atmospheric turbulence. The long term OTF results from simply taking an image of sufficiently long term which sees effectively all possible turbulence configurations.

When observing the effect of diffraction as the result of the finite aperture of a point source, it can be seen that the resulting image is not uniform. Considering the point spread function to have the same shape in the image plane regardless of its position, the image function is the result of the convolution of the source function and the

optics diffraction function. The convolution theorem, as described in Fourier transform theory, yields

$$i(v(x),v(y))=H(v(x),v(y))*o(v(x),v(y))*M(v(x),v(y)) \quad (2.1)$$

where

$i(v(x),v(y))$ = Fourier transform of the image function

$H(v(x),v(y))$ = Fourier transform of the optics diffraction function

$o(v(x),v(y))$ = Fourier transform of the object function

$M(v(x),(v(y))$ = Modulation Transfer Function of the atmosphere

v = spatial frequency

A point source such as a laser can be analyzed in two dimensions using a point spread function. Through Fourier transform theory, the image point spread function is transformed into a two dimensional OTF of the optical signal. This problem may be simplified by scanning the image point spread function using a vidicon or slit-scanning system. In applying the convolution theorem, the Fourier transform of the image point spread function is multiplied with the Fourier transform of the optical system resulting in the Fourier transform of the overall system. The Abel

transform described by Griem, is applied to this result to re-transform the one-dimensional image LSF into a two dimensional image PSF [Ref. 7].

As demonstrated by Crittenden, and others, a numerical value for C_n^2 may be obtained by curve fitting using the following:

$$M = \exp(-21.49 * C_n^2 * Z * f^{5/3} * \lambda^{-1/3}) \qquad (2.2)$$

where

M = MTF of the atmosphere

Z = Range in meters

f = F*v = Angular spatial frequency in cycles/radian

F = focal length of the optical system

v = linear spatial frequency in cycles/meter

λ = wavelength in meters

C_n^2 is obtained by a linear regression of

$$\ln(M) = -21.49 * C_n^2 * Z * f^{5/3} * \lambda^{-1/3} \qquad (2.3)$$

where C_n^2 is the only parameter.

III. EXPERIMENTAL PROCEDURE

The experiment is performed using two different laser sources, helium-neon (He-Ne) and gallium-arsenide (Ga-As). Figure 3.1 shows a block diagram of the experimental set-up used for both lasers.

The measurement of C_n^2 along the optical path is made by using a vidicon and telescope at the far end of the corridor in the basement of Spanagel Hall. Atmospheric turbulence is produced by nine overhead hot air ventilators. The optical equipment similar to that described in reference 3, consists of a 6 inch diameter Cassegrain telescope with a 90 inch focal length [Ref. 3].

Using neutral density absorption filters to attenuate the intensity of the laser beam, the telescope is illuminated at its input aperture. The light image from the telescope is split by a beam splitter with one beam sent to the vidicon and the other to a slit scanner for comparison of the two systems. The light is then transformed to an analog signal and recorded on a Panasonic(NV-9240) video tape recorder for processing. The data from the slit scanning system are recorded on a precision instrument tape recorder for analysis. After recording the video signal,

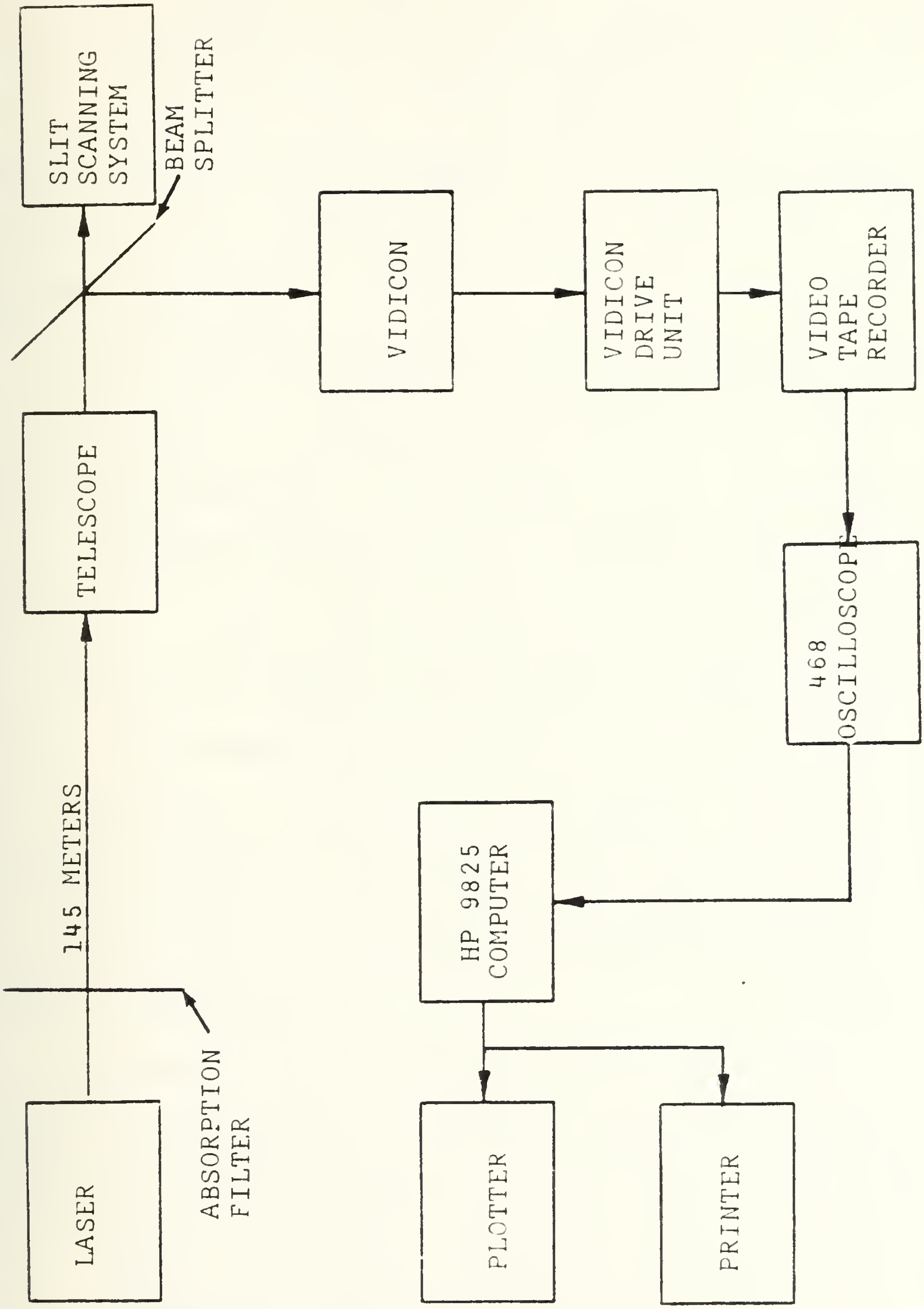

Fig. 3.1 Laser Experimental Set-Up

analysis is accomplished using a Tektronix 468 Digital Storage Oscilloscope and a Hewlett Packard 9825 Computing System. The linearity of the Panasonic tape recorder was demonstrated in Crager's thesis. This is also confirmed by observing the images as real time and recorded displays [Ref. 3].

The 468 oscilloscope is set-up in the following manner. An input signal to channel A is located on the scope using the non-storage mode. The signal is then displayed using the A INTENSITY switch of the horizontal display. This intensified zone is used to position the B sweep (delayed) to the desired location within the A sweep interval to obtain an expanded view of a waveform for examination. Once the waveform is centered on the scope, the horizontal display is switched to B DLY'D. This is done to facilitate the digital storage circuitry time base by using the setting of the B TIME/DIV switch.

The waveform is expanded in time by decreasing the A TIME/DIV switch setting and moving the waveform back to the middle of the scope using the delay time position control dial. When a representative waveform is obtained, the B TIME/DIV switch is then used in conjunction with the delay

time position control dial to expand the time scale until the single central maximum of the waveform is centered and one horizontal sweep is displayed.

Once this waveform is satisfactorily obtained, the AVG storage mode is selected. The 468 will average the input signal for a selected number of sweeps and display the accumulated waveform. All data for this thesis use 32 sweeps for each average. The 468 is now ready to transfer data when interrogated by the HP 9825.

The computer is the controller for all interfacing operations with only a minimal amount of operator interaction. The operator interface is mainly to ensure that the equipment is properly set up and to select if data are to be plotted. Digitization of the analog signal is accomplished by the Tektronix 468. Processed waveform data are transferred from the microprocessor memory to the Storage Display RAM [Ref. 8].

The controlling program of the HP 9825 interrogates the 468 via the IEEE 488 interface bus. The 468 receives the data request from the controller and sends the waveform message, both preamble and data. The waveform message is stored in the calculator memory for further processing.

When the message is completed, the 468 concludes with an end of instruction terminator and the controller takes control of the bus again.

Data processing of the digitized signal begins with the data being stored on magnetic tape for further use by the computer. The data are stored, processed and plotted by the main program. Subroutines are called as necessary for their specific uses. The two signal waveforms are recorded and processed. First, the signal from a laser beam incident at the aperture of the telescope is recorded. Figures 3.2 and 3.3 show these recorded data which are referred to as the source data. Second, for calibration, the signal from a laser beam with a diffraction grating in place at the aperture of the telescope is recorded. The entire system is calibrated by using a grating that consists of closely spaced vertical bars in front of the telescope. This produces a diffraction pattern in the image plane and allows for the calculation of the scale factor. These recorded data are referred to as the scale data and are plotted in Figures 3.4 and 3.5.

During the recording process, the signal from the laser beam is recorded on video tape while the signal is observed

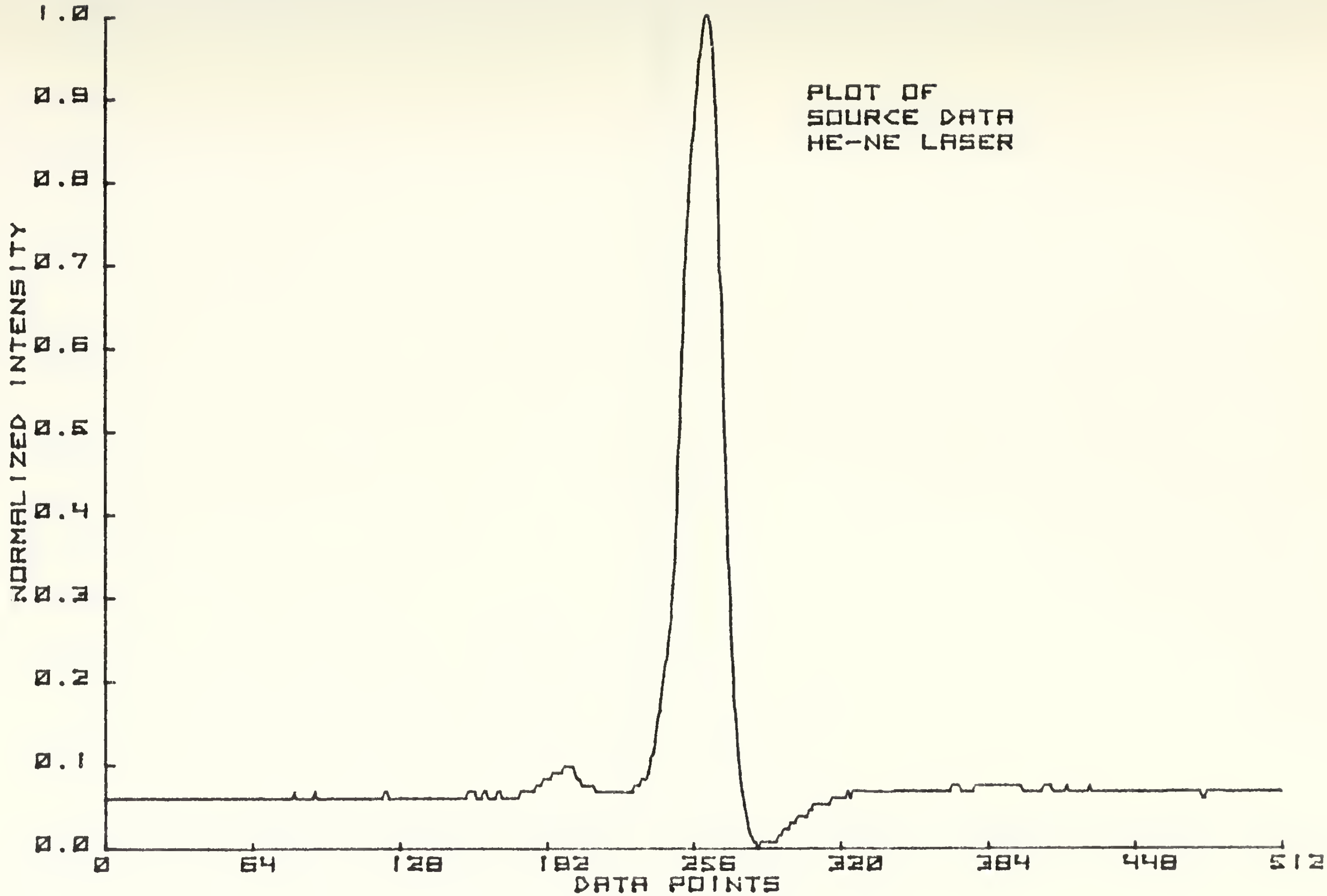

Fig. 3.2 Plot of Source Data for He-Ne Laser

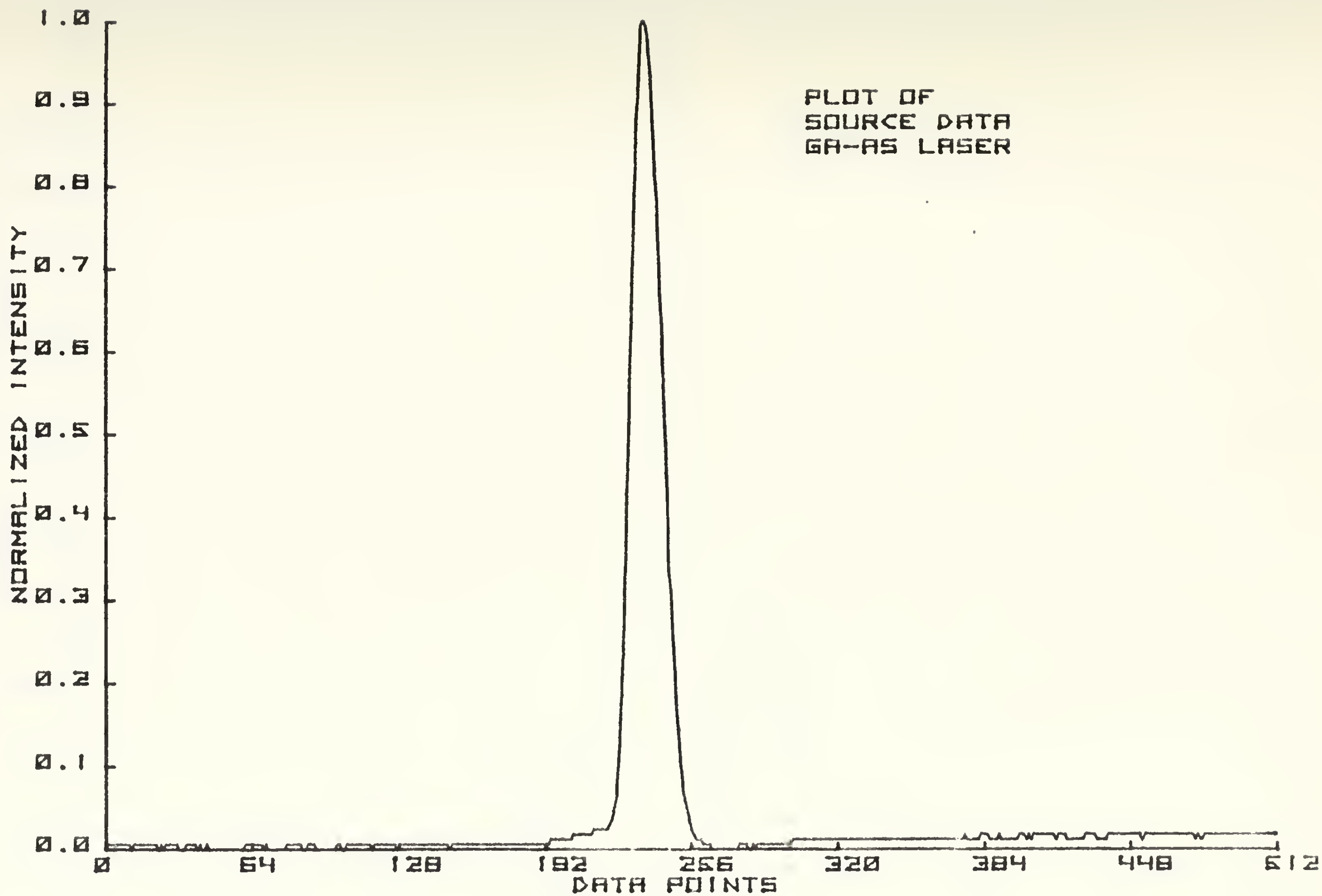

Fig. 3.3 Plot of Source Data for Ga-As Laser

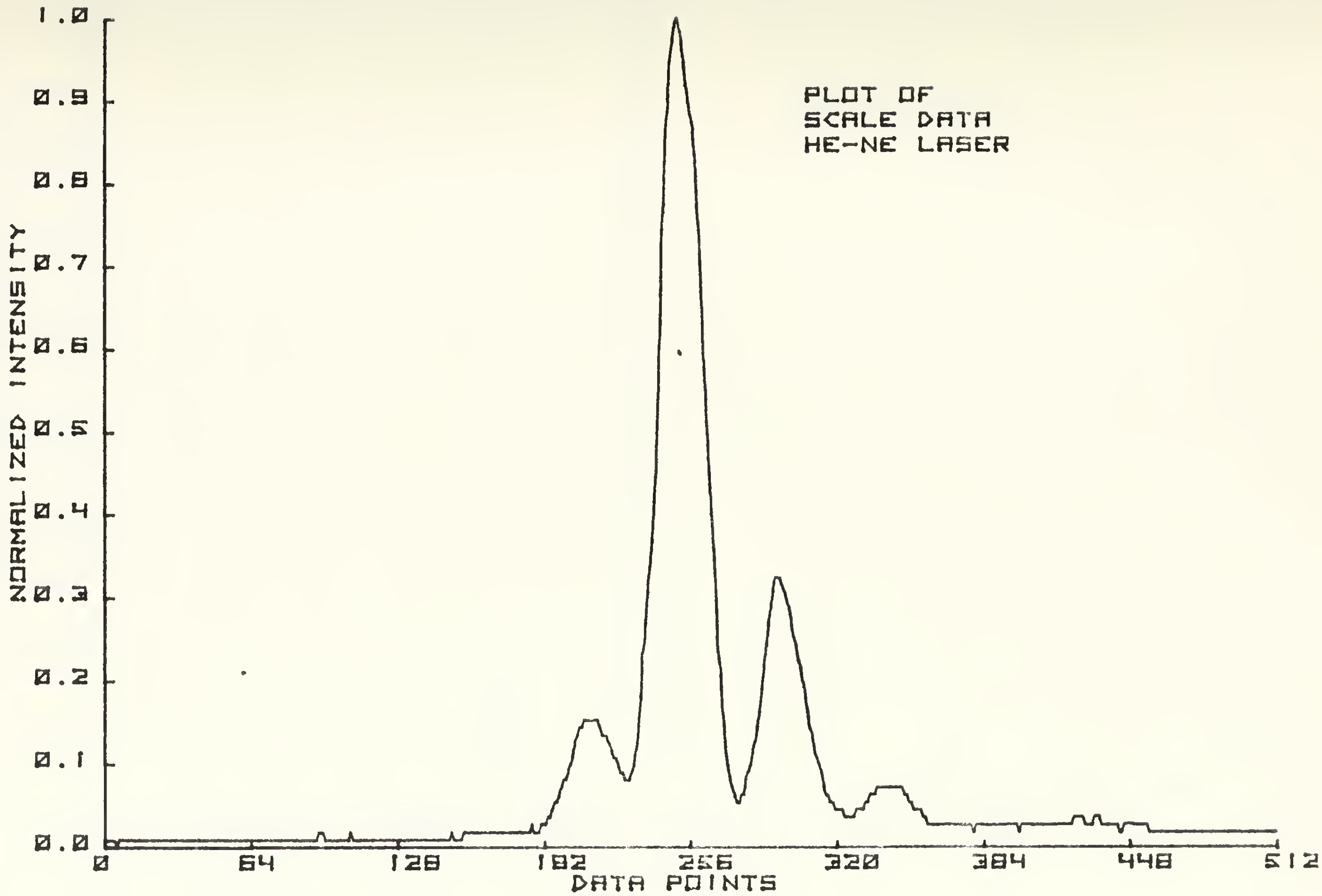

Fig. 3.4 Plot of Scale Data for He-Ne Laser

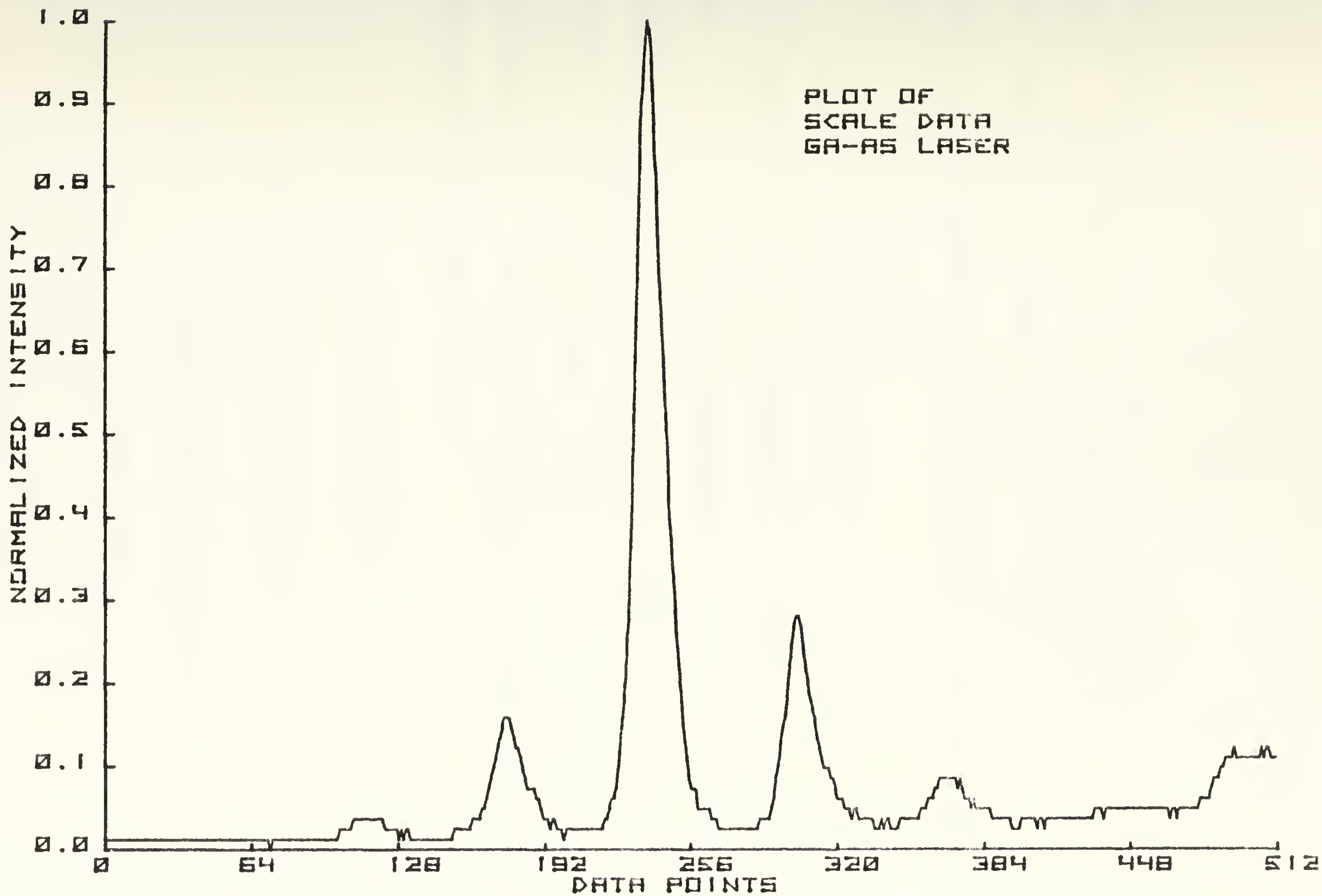

Fig. 3.5 Plot of Scale Data for Ga-As Laser

on the 468 oscilloscope. This procedure is important in data taking since laser alignment and telescope focusing are often very sensitive to minor movements of either. Upon completion of recording the data on video tape, the experiment is concerned with the compilation and analysis of data. The first item to be determined is the scale factor. Once the scale factor has been determined, it will remain constant throughout the calculations, unless the telescope focal length is changed.

The scale factor is calculated in the following manner. The number of points between the peaks in the plot of the scale data (diffraction grating in place in front of the telescope aperture) is measured. When the distance between peaks, the spacing between the bars in the grating, and the wavelength of the laser are known, the scale factor can be calculated from the relation

$$\sin(\text{theta}) = \lambda / d \tag{3.1}$$

$$\sin(\text{theta}) \sim \text{theta} \tag{3.2}$$

(small angle approximation)

$$sf = (\text{theta}) / ndp \tag{3.3}$$

$$sf = \lambda / (d * ndp) \tag{3.4}$$

where

sf = scale factor in radians per point

λ = wavelength in meters

d = spacing between lines on grating in meters

ndp = number of points between central maximum and first order diffraction peak

This standardizes the data for the abscissa of the plots in radians per point.

The program now takes the transferred data and computes a point spread function as shown in Figures 3.6 and 3.7. Figures 3.8 and 3.9 represent the point spread function after integration using equation (1.1) to obtain a line spread function. Next, the Fourier transform of the line spread function is calculated. These curves are plotted in Figures 3.10 and 3.11.

The diffraction of the optics is computed and plotted as shown in Figures 3.12 and 3.13. Following the same method as before, the line spread function of the optics function is calculated and plotted in Figures 3.14 and 3.15. The program now takes the Fourier transform of the optics function LSF. These data are plotted in Figures 3.16 and

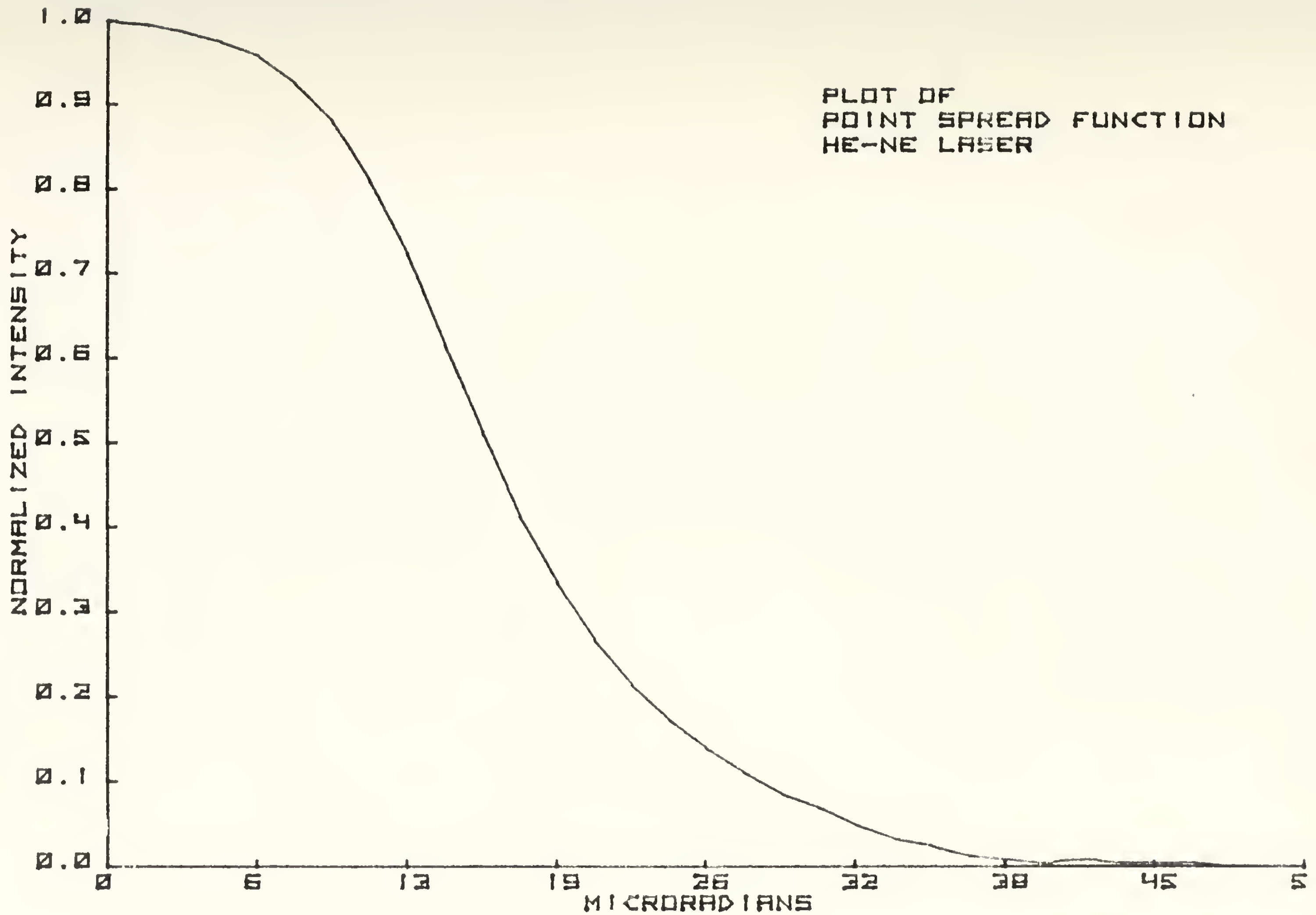

Fig. 3.6 Plot of Point Spread Function for He-Ne Laser

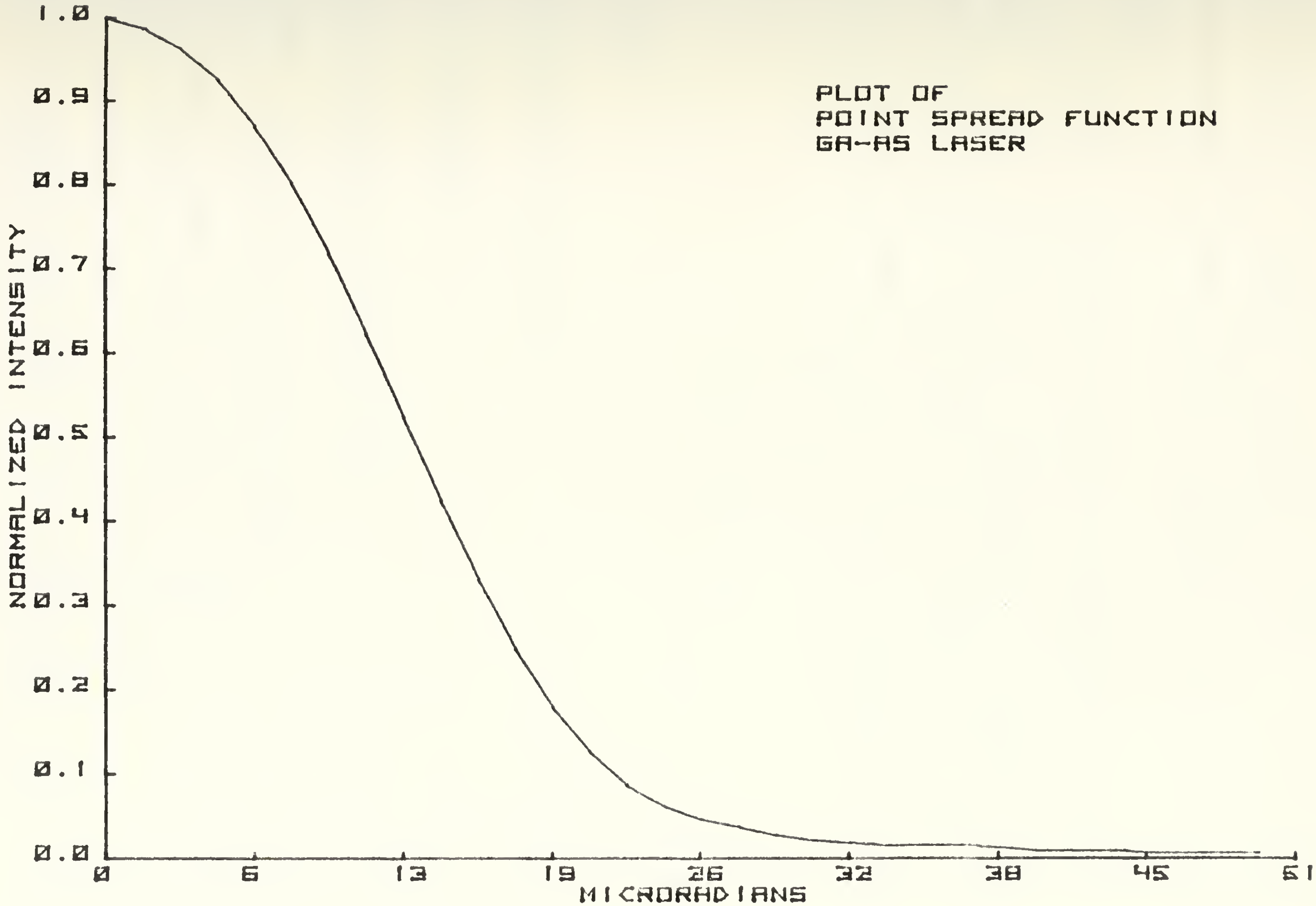

Fig. 3.7 Plot of Point Spread Function for Ga-As Laser

3.17. The Fourier transforms of the system and optics are divided point by point to yield the MTF of the atmosphere, as shown in Figures 3.18 and 3.19. The MTF of the atmosphere will be used to calculate the long-term value of C_n^2.

C_n^2 is calculated by linear regression of ln(MTF) versus $f^{5/3}$. The slope of the regression is proportional to C_n^2. Two simultaneous equations are solved using Cramer's rule and the matrix ROM of the HP 9825. The equations used are

$$a*X + b*I = Y \tag{3.5}$$

$$a*X^2 + b*X = Y*X \tag{3.6}$$

where

a = slope of the curve

b = intercept

X = summation of (I*sf)**(5/3)

X**2 = summation of (I*sf)**(10/3)

I = point number (total number = 256)

Y = summation of natural logarithm of MTF

Y*X = summation of products of two values

C_n^2 is obtained from the above information using equation (2.3). This will yield

$$C_n^2 = a/(-21.49*Z*f**(5/3)* **(-1/3)) \qquad (3.7)$$

The program now goes to the prediction phase after calculating C_n^2 for the atmosphere. If a Gaussian distribution for an input source and a value of the standard deviation are assumed or known, the resulting source function can be calculated. A plot of the computed source is shown in Figure 3.20. It can be seen from this plot that the half-width at half-maximum is approximately 4 microradians. As before, the line spread function of the computed source is calculated and plotted. This is shown in Figure 3.21. The Fourier transform of this data is calculated and plotted in Figure 3.22.

The program now multiplies the Fourier transform of the computed source with the transform of the system (including the atmosphere) and plots the result in Figures 3.23 and 3.24. Plots of the inverse Fourier transform are shown in Figures 3.25 and 3.26. Next, the Abel transform is computed and plotted in Figures 3.27 and 3.28. The Abel transform, as described in Chapter II, transforms a one-dimensional

line spread function into a two-dimensional point spread function. Finally, the fraction of power inside a circle of radius R is calculated and plotted as shown in Figures 3.29 and 3.30. This is the fraction of power that one would expect to be incident on a target using the measured value of atmospheric turbulence as an input parameter. A detailed analysis of the computer program is the subject of the next chapter.

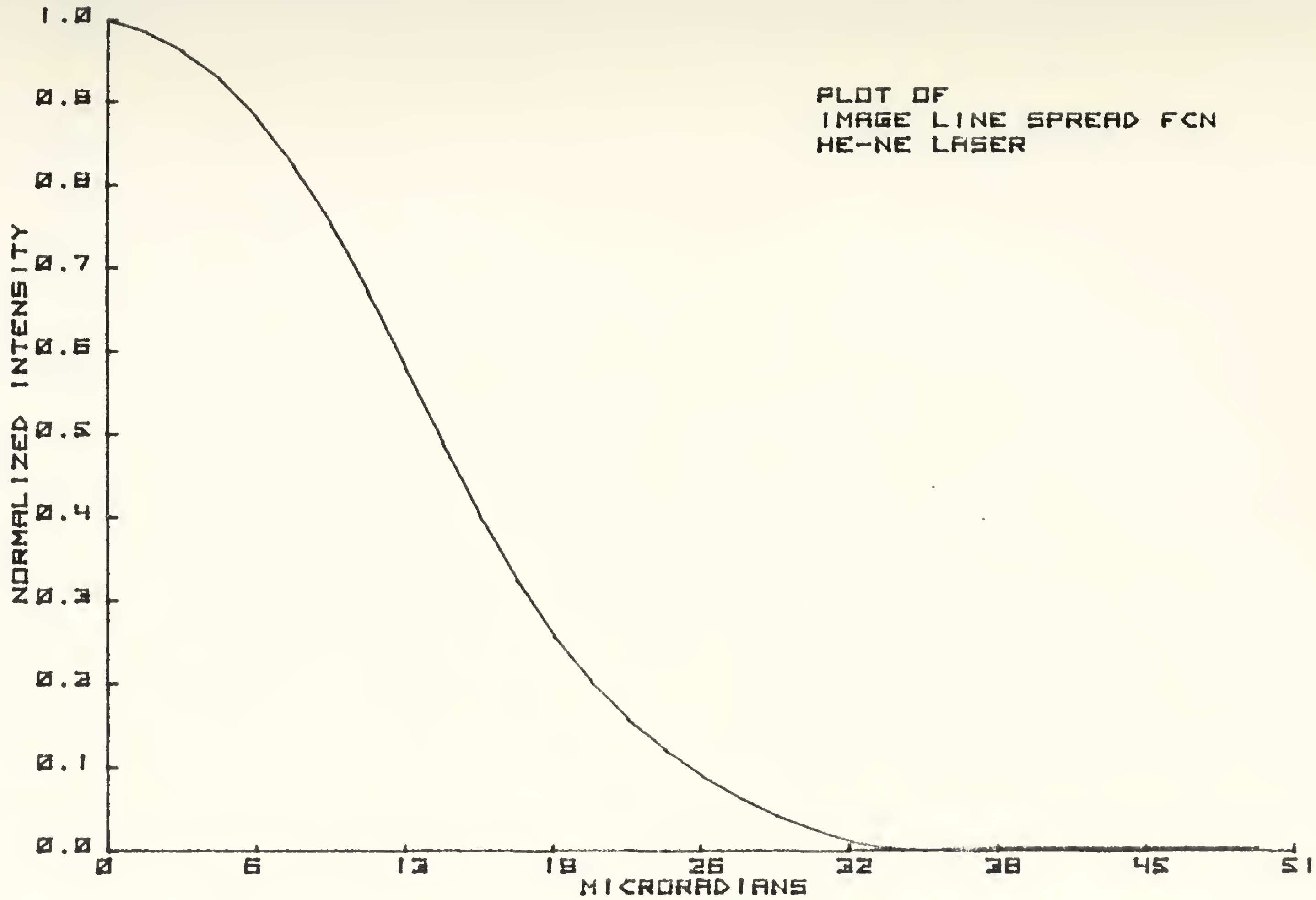

Fig. 3.8 Plot of Line Spread Function for He-Ne Laser

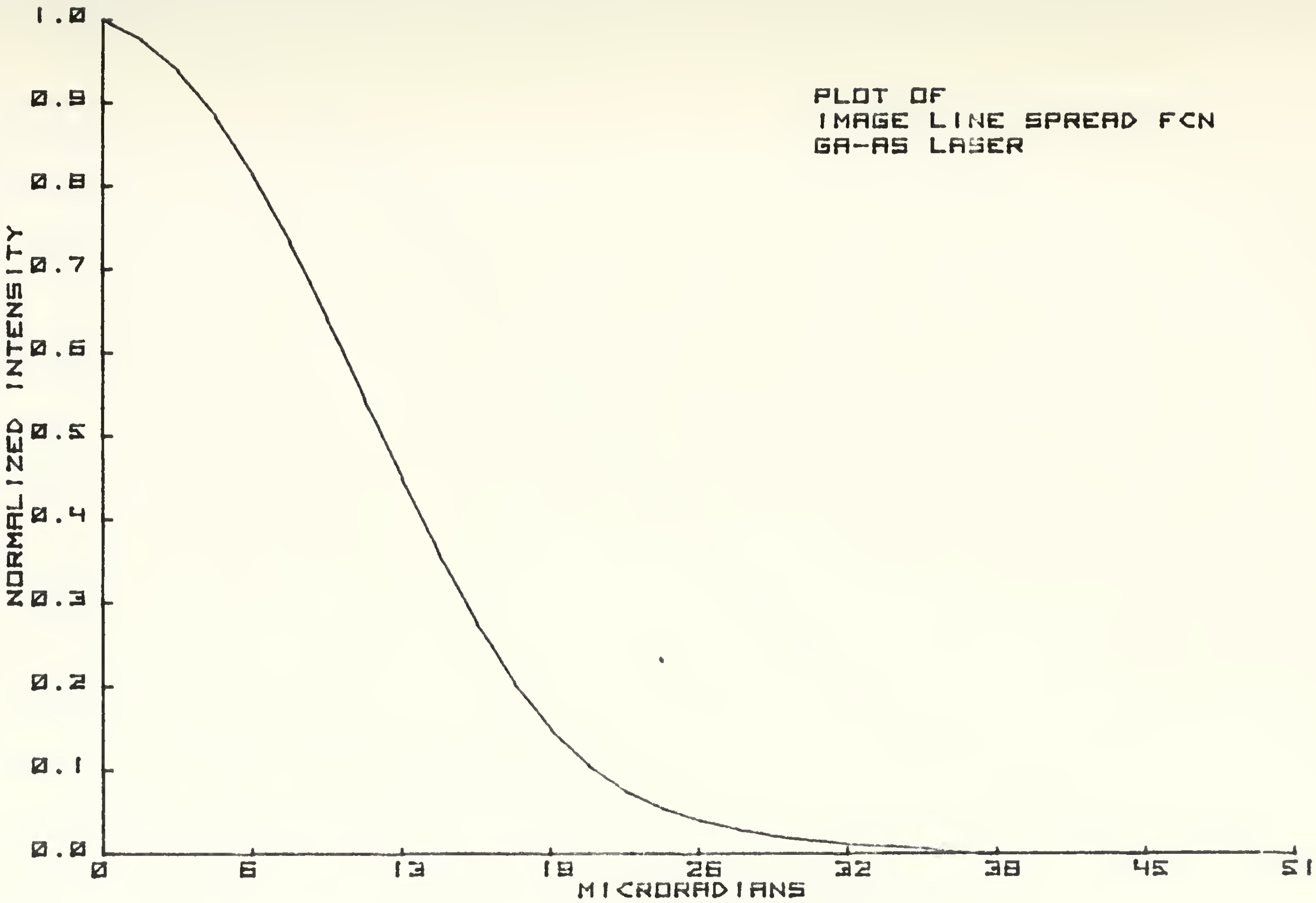

Fig. 3.9 Plot of Line Spread Function for Ga-As Laser

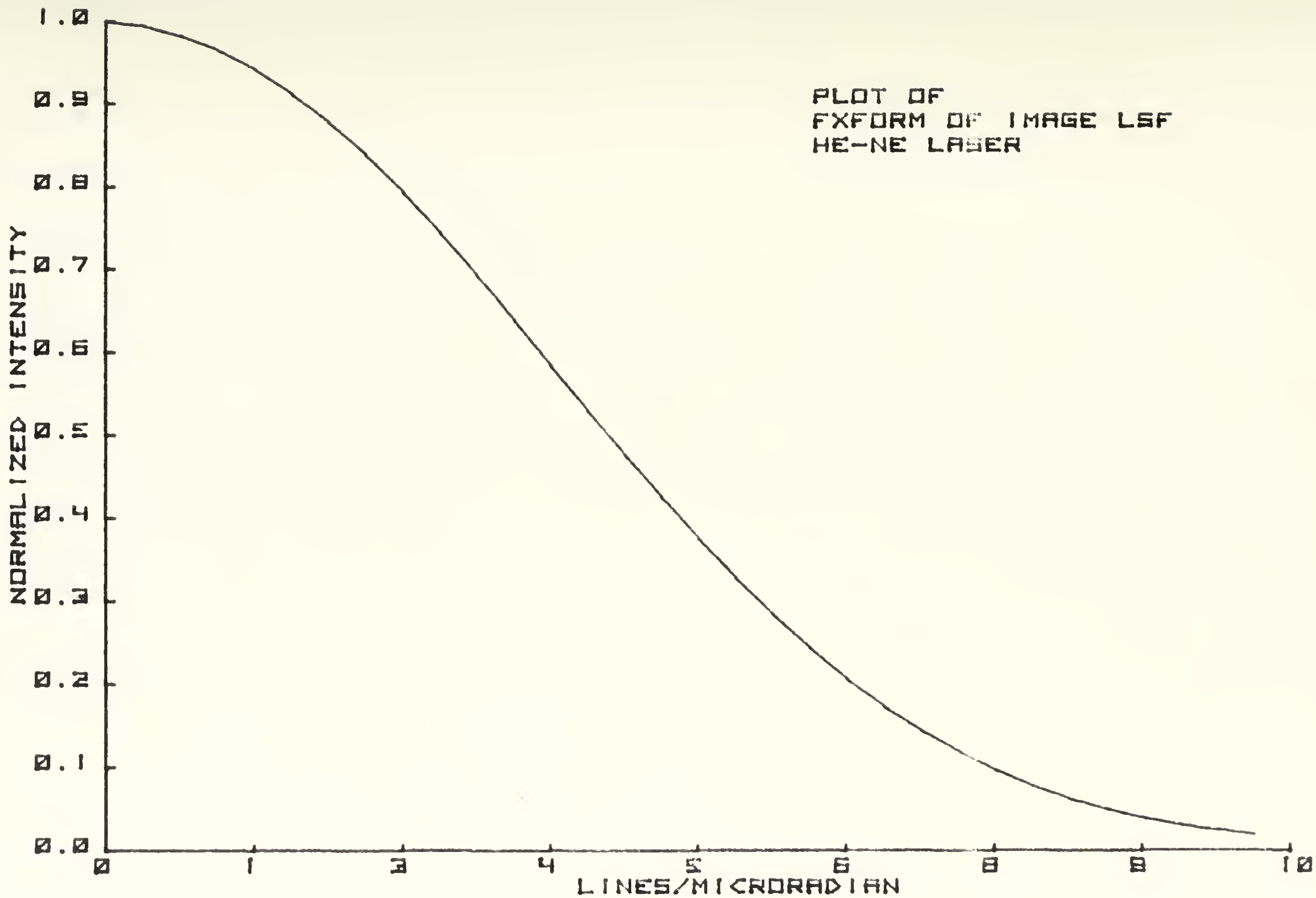

Fig. 3.10 Plot of Fourier Transform of LSF for He-Ne Laser

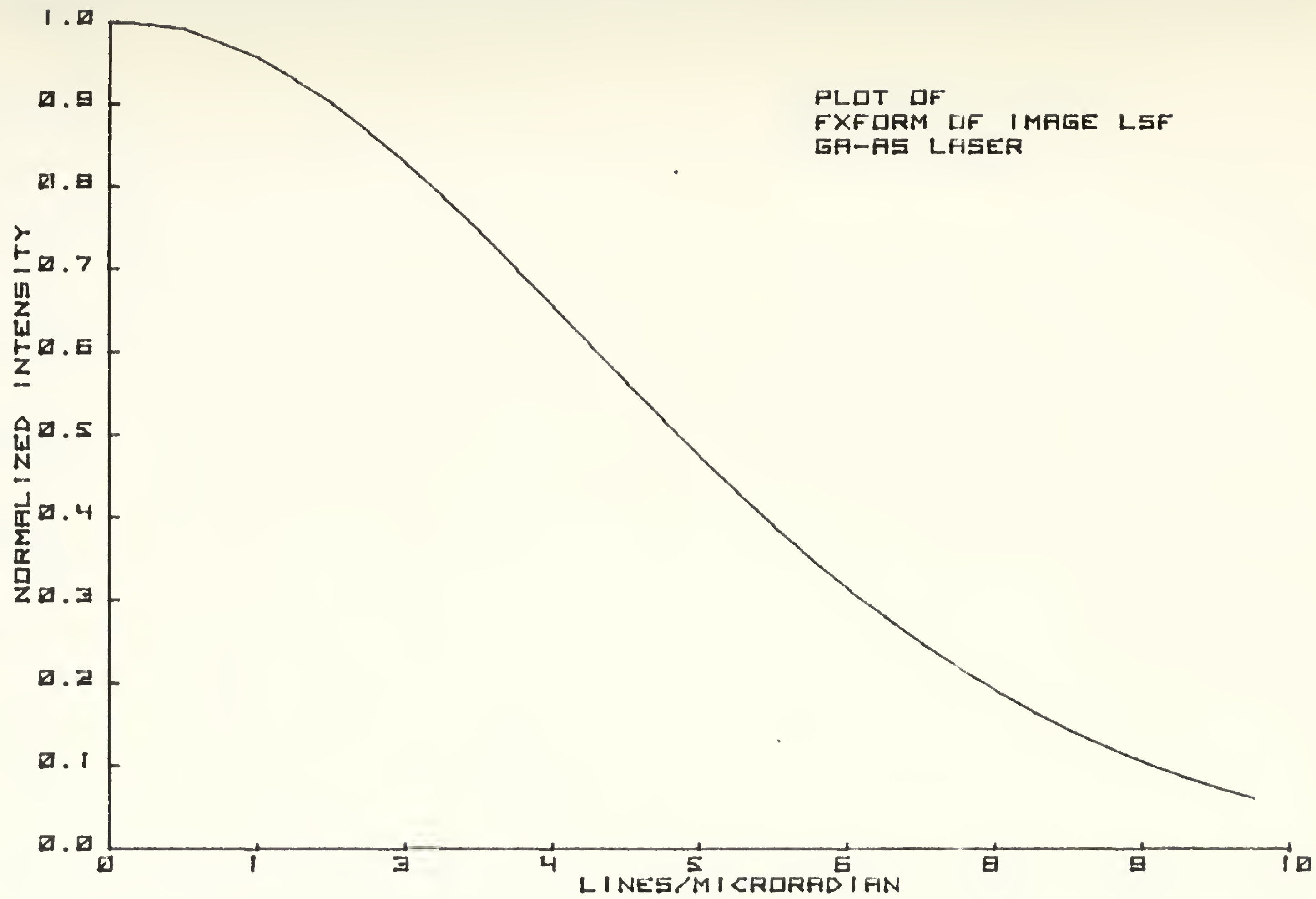

Fig. 3.11 Plot of Fourier Transform of LSF for Ga-As Laser

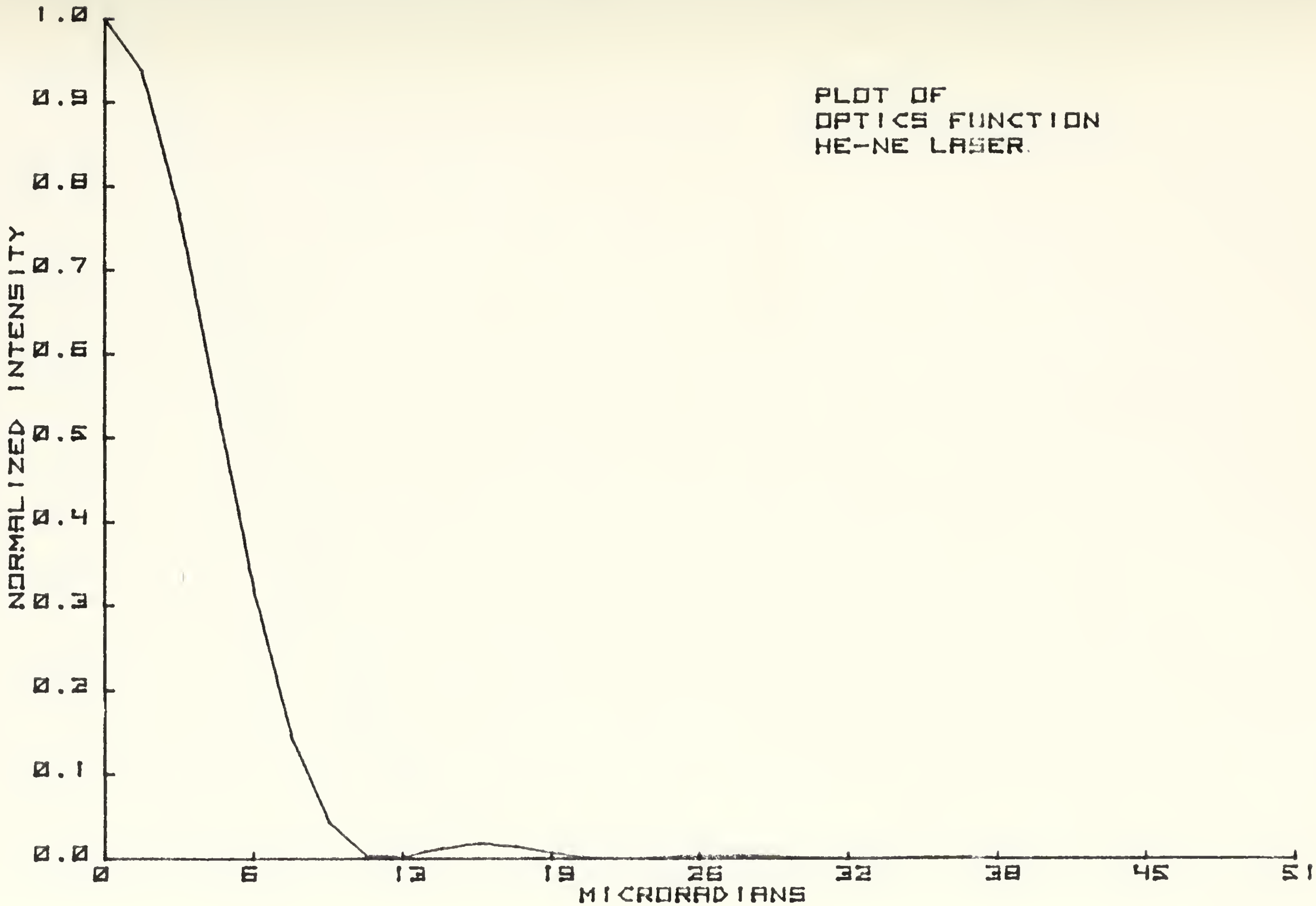

Fig. 3.12 Plot of Optics Diffraction for He-Ne Laser

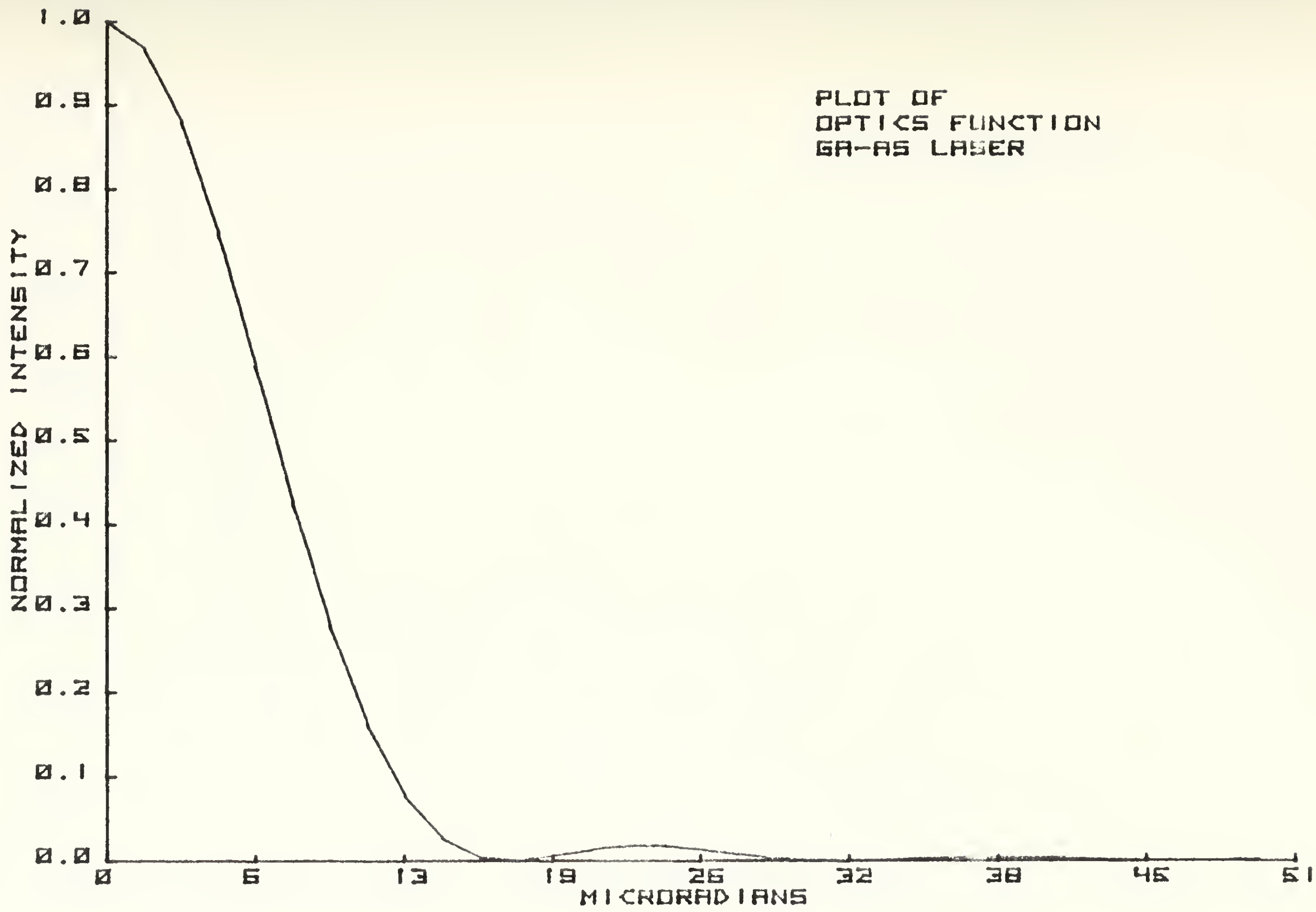

Fig. 3.13 Plot of Optics Diffraction for Ga-As Laser

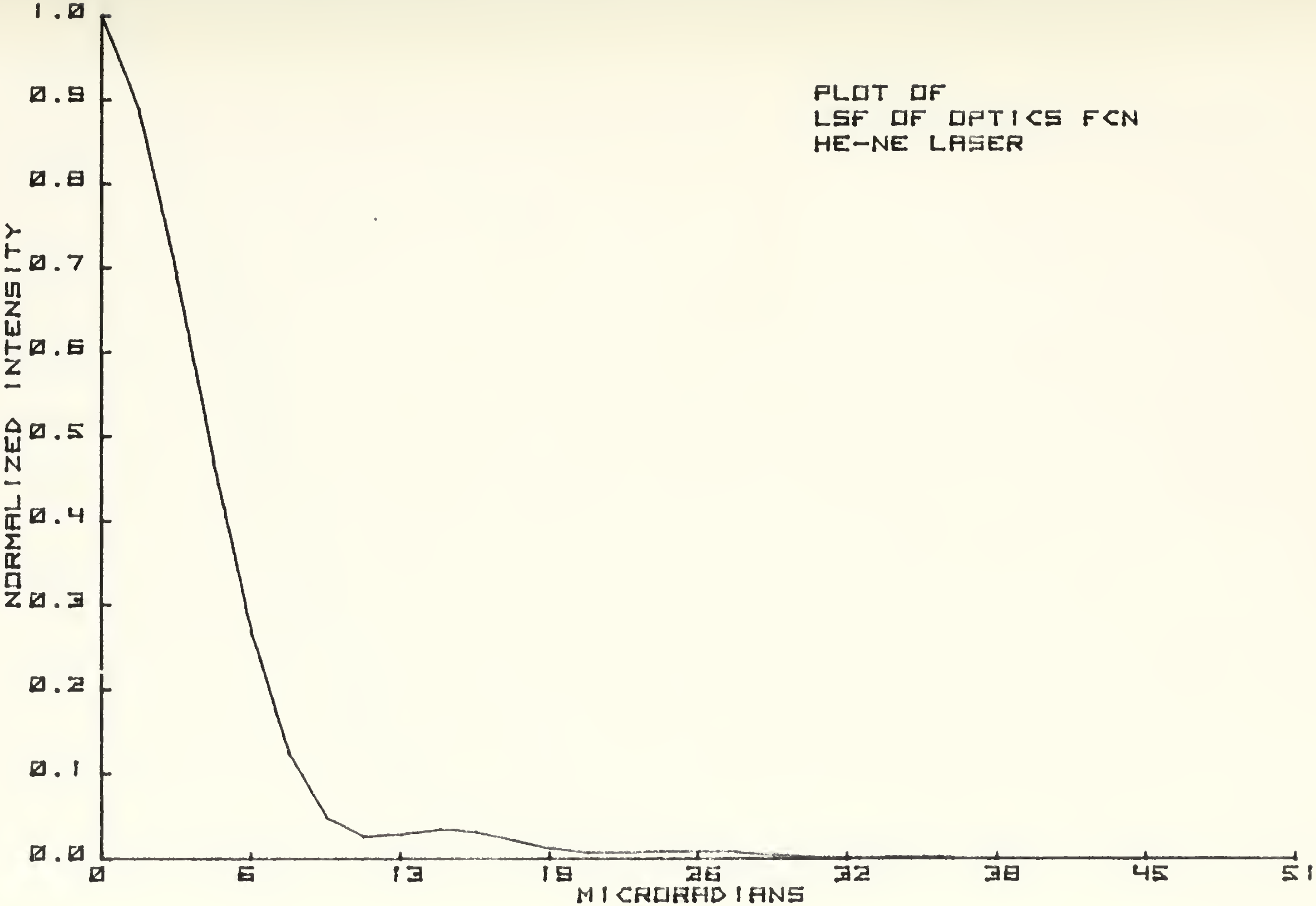

Fig. 3.14 Plot of LSF of Optics Function for He-Ne Laser

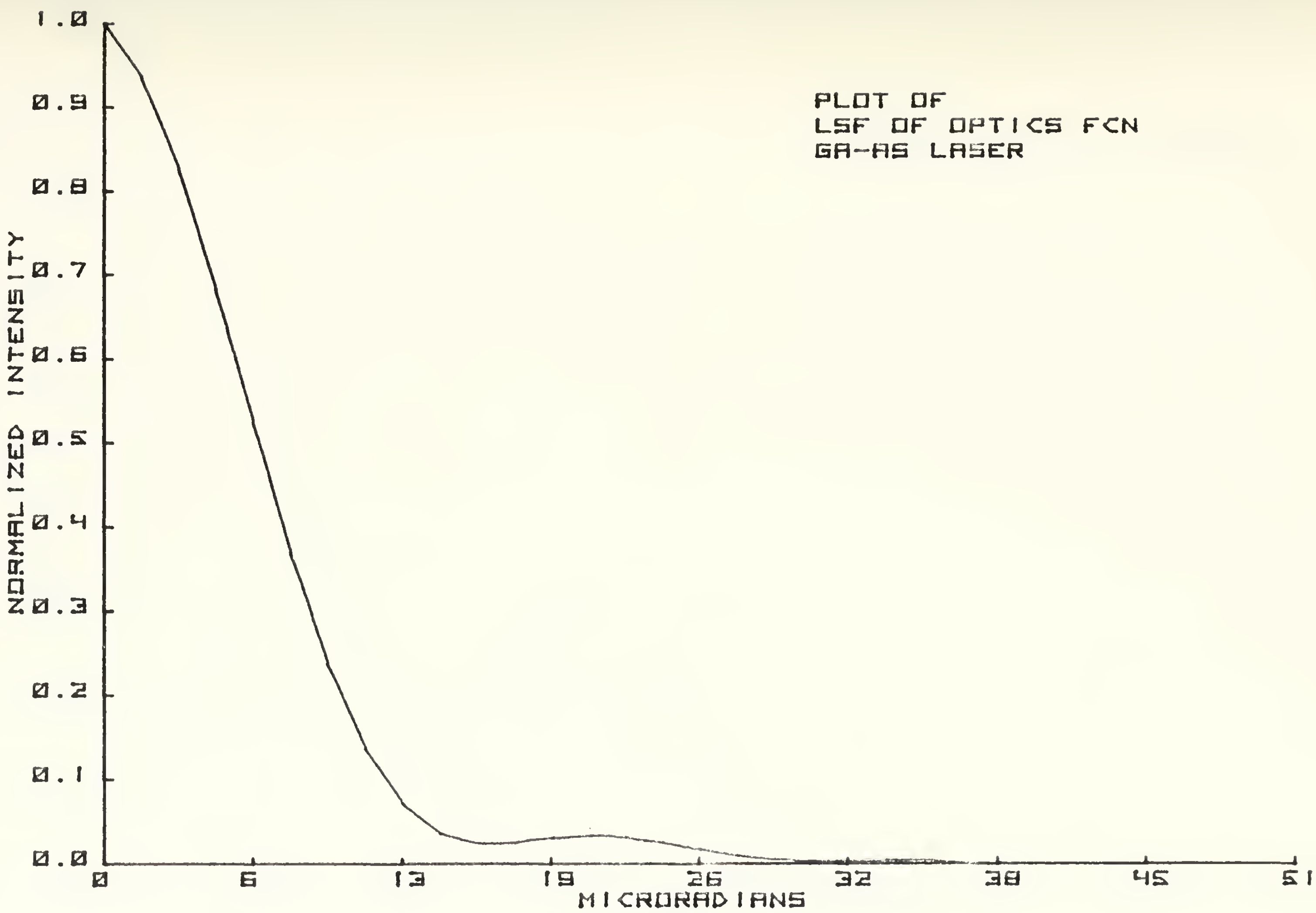

Fig. 3.15 Plot of LSF of Optics Function for Ga-As Laser

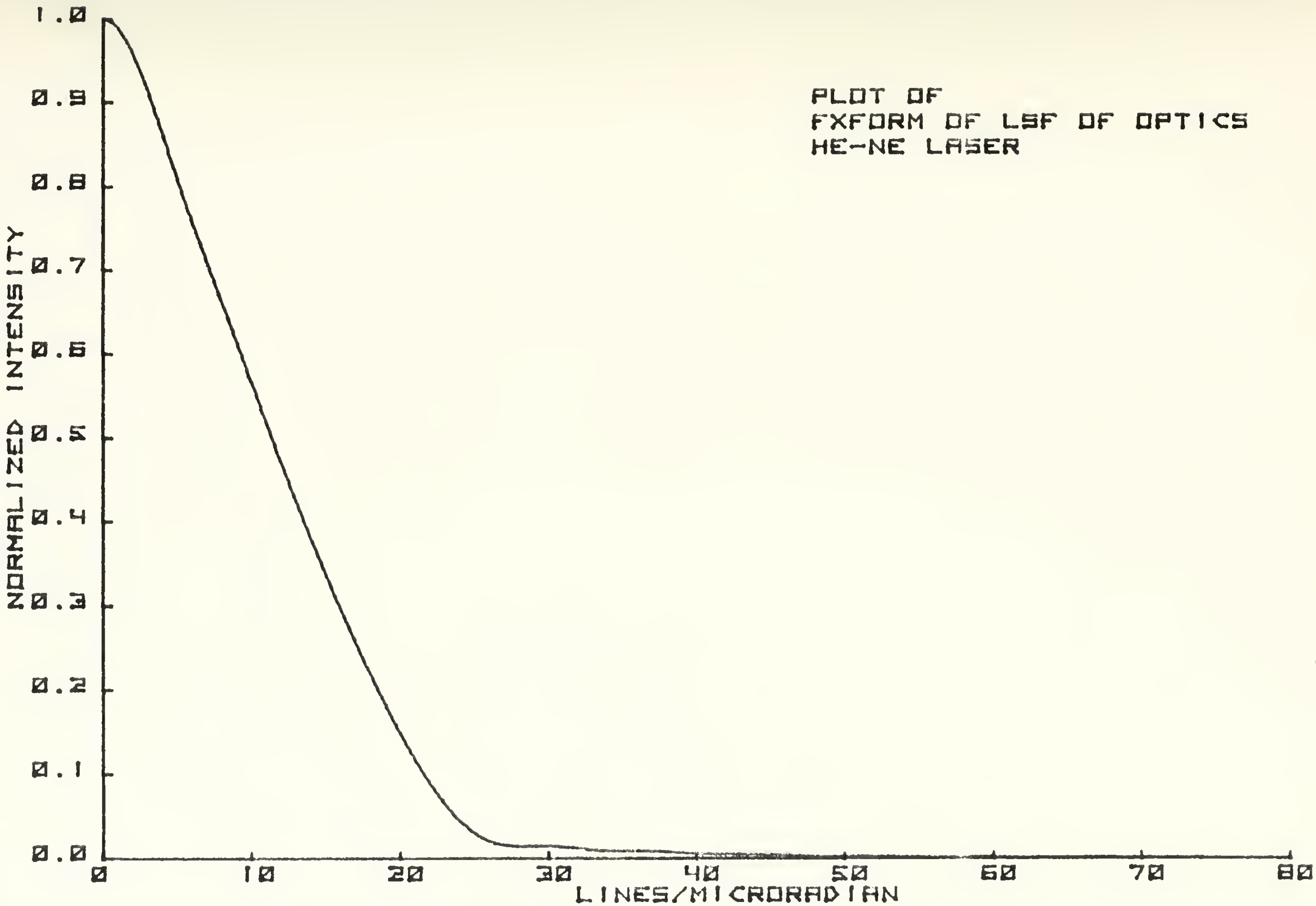

Fig. 3.18 Plot of Fourier Transform of LSF of Optics Function for He-Ne Laser

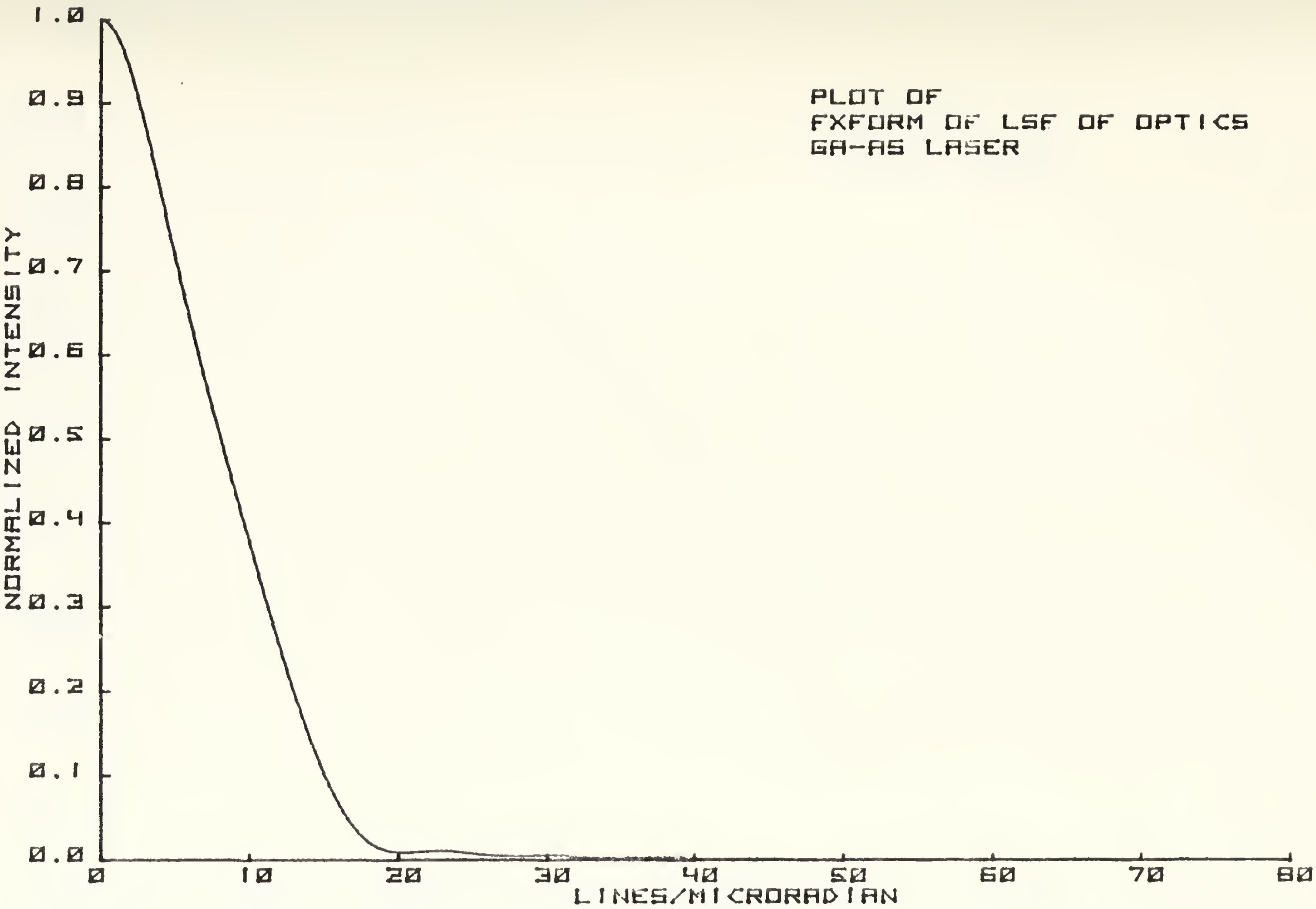

Fig. 3.17 Plot of Fourier Transform of LSF of Optics Function for Ga-As Laser

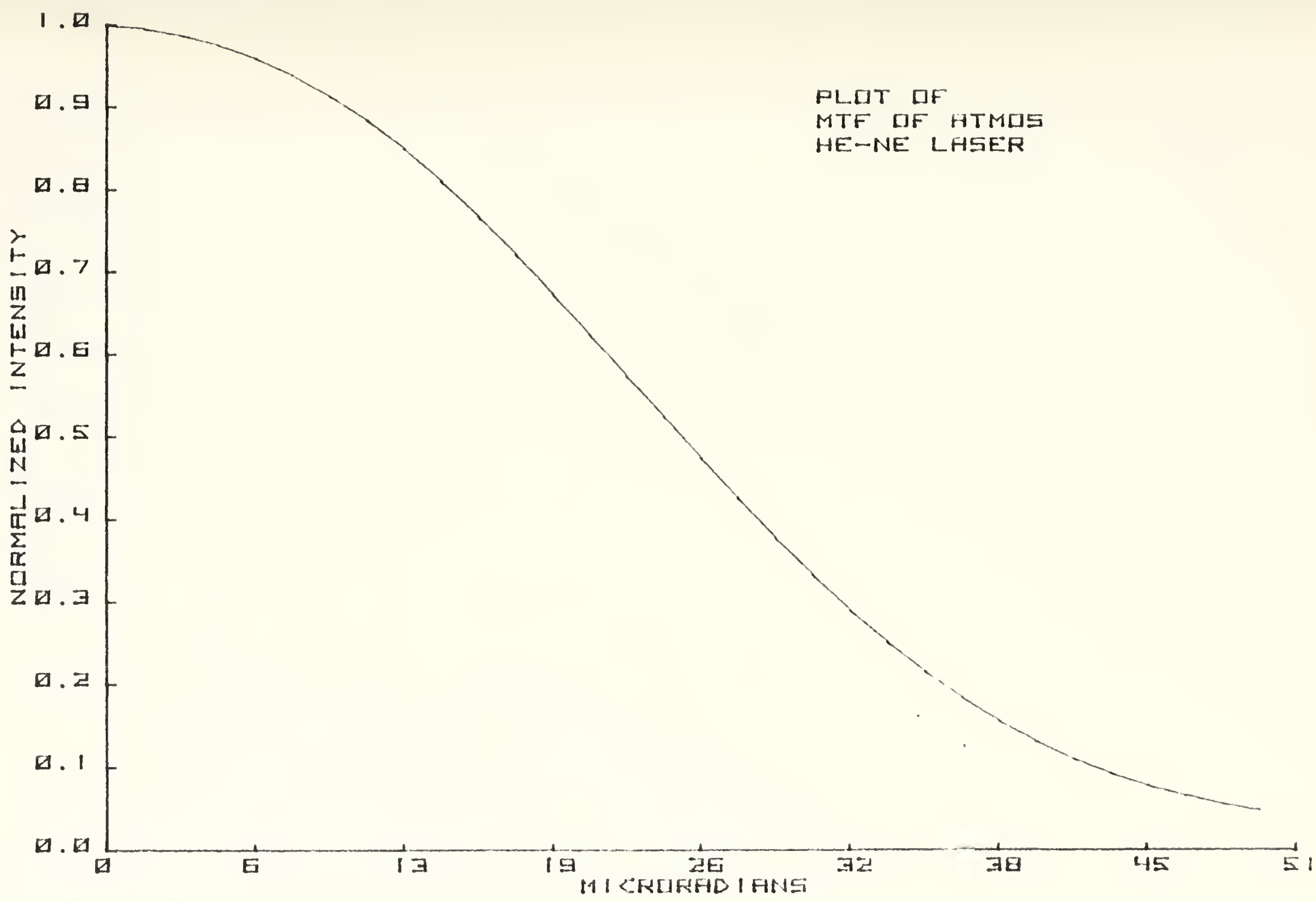

Fig. 3.18 Plot of MTF of Atmosphere for He-Ne Laser

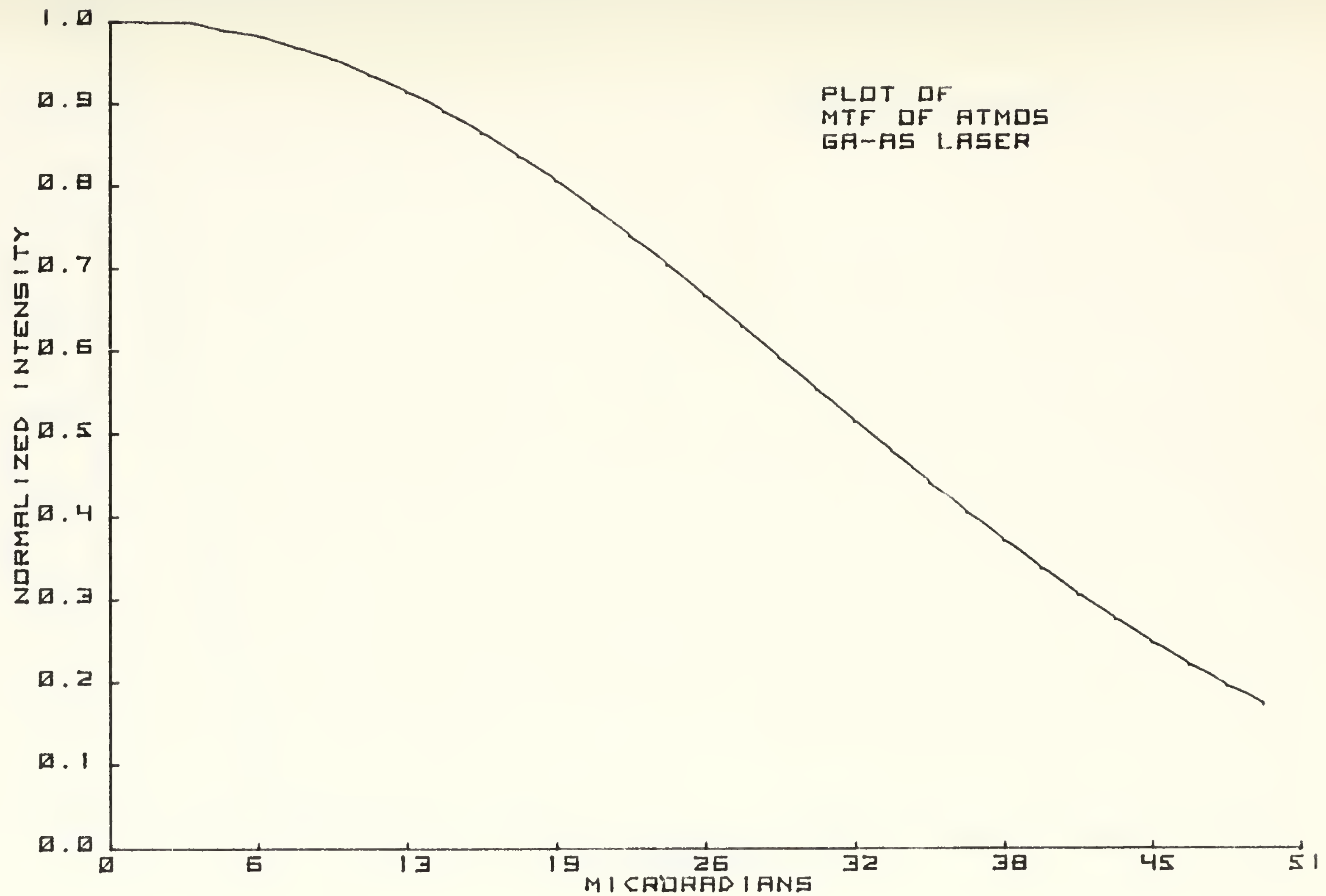

Fig. 3.19 Plot of MTF of Atmosphere for Ga-As Laser

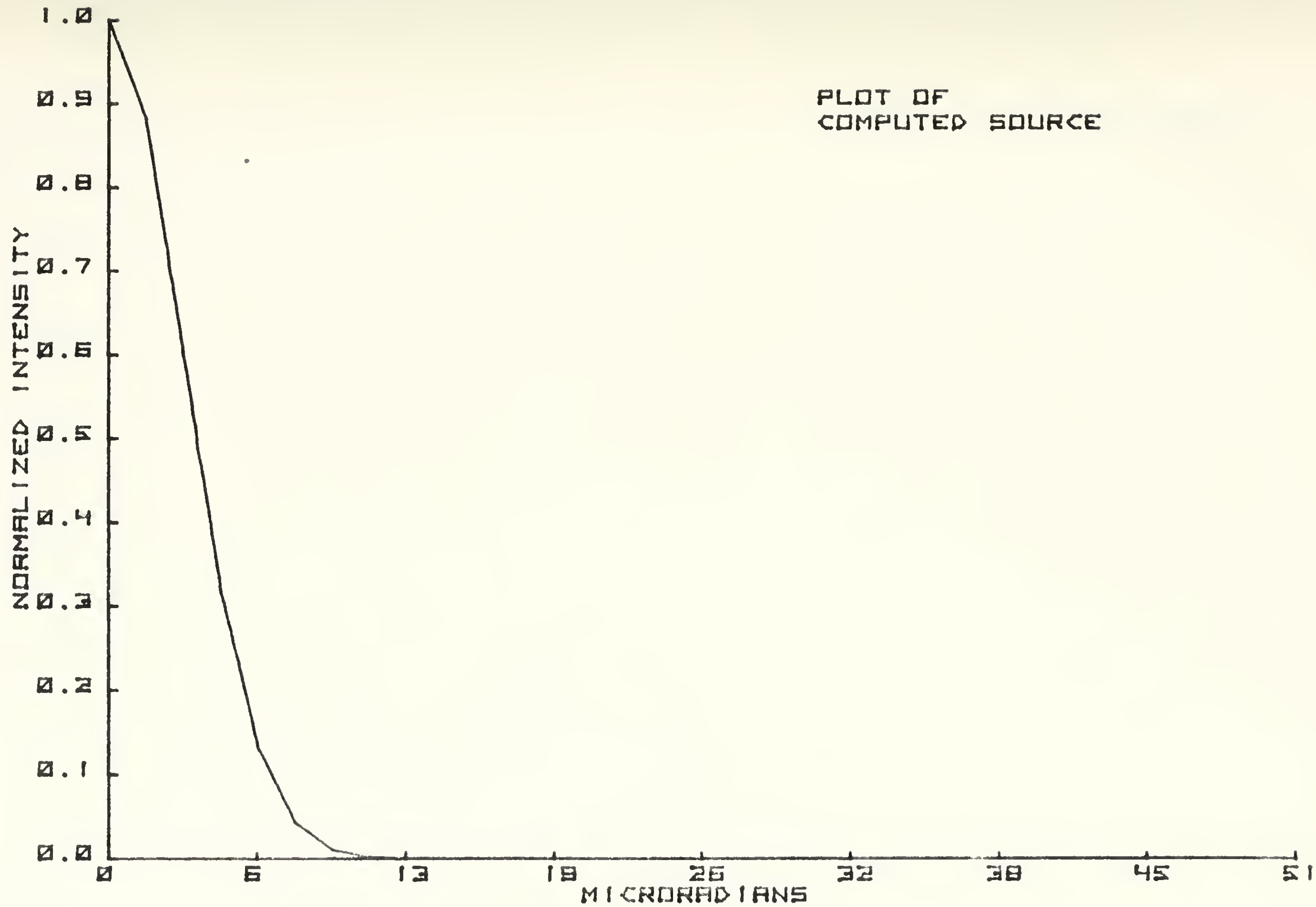

Fig. 3.20 Plot of Computed Source

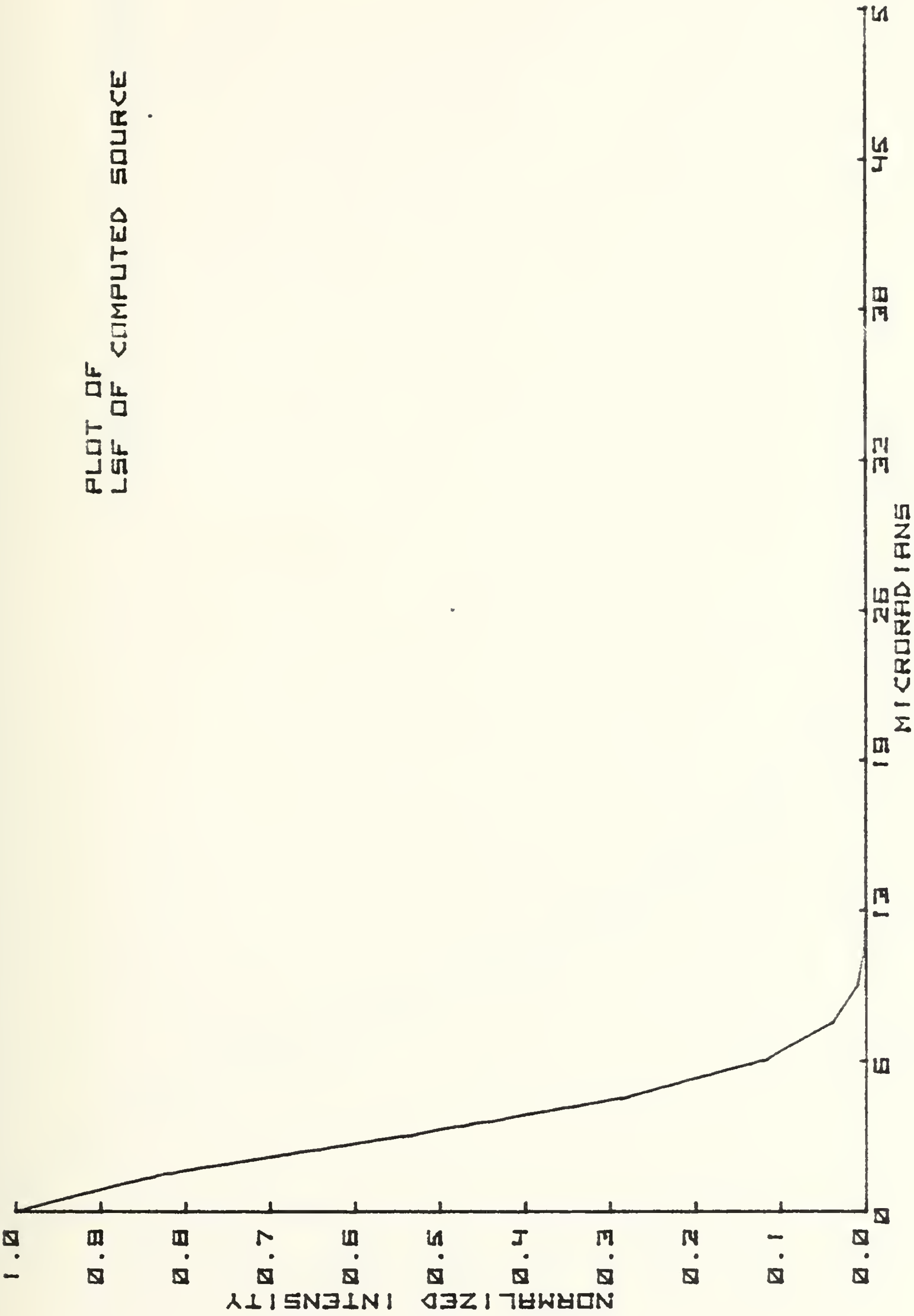

Fig. 3.21 Plot of Line Spread Function of Computed Source

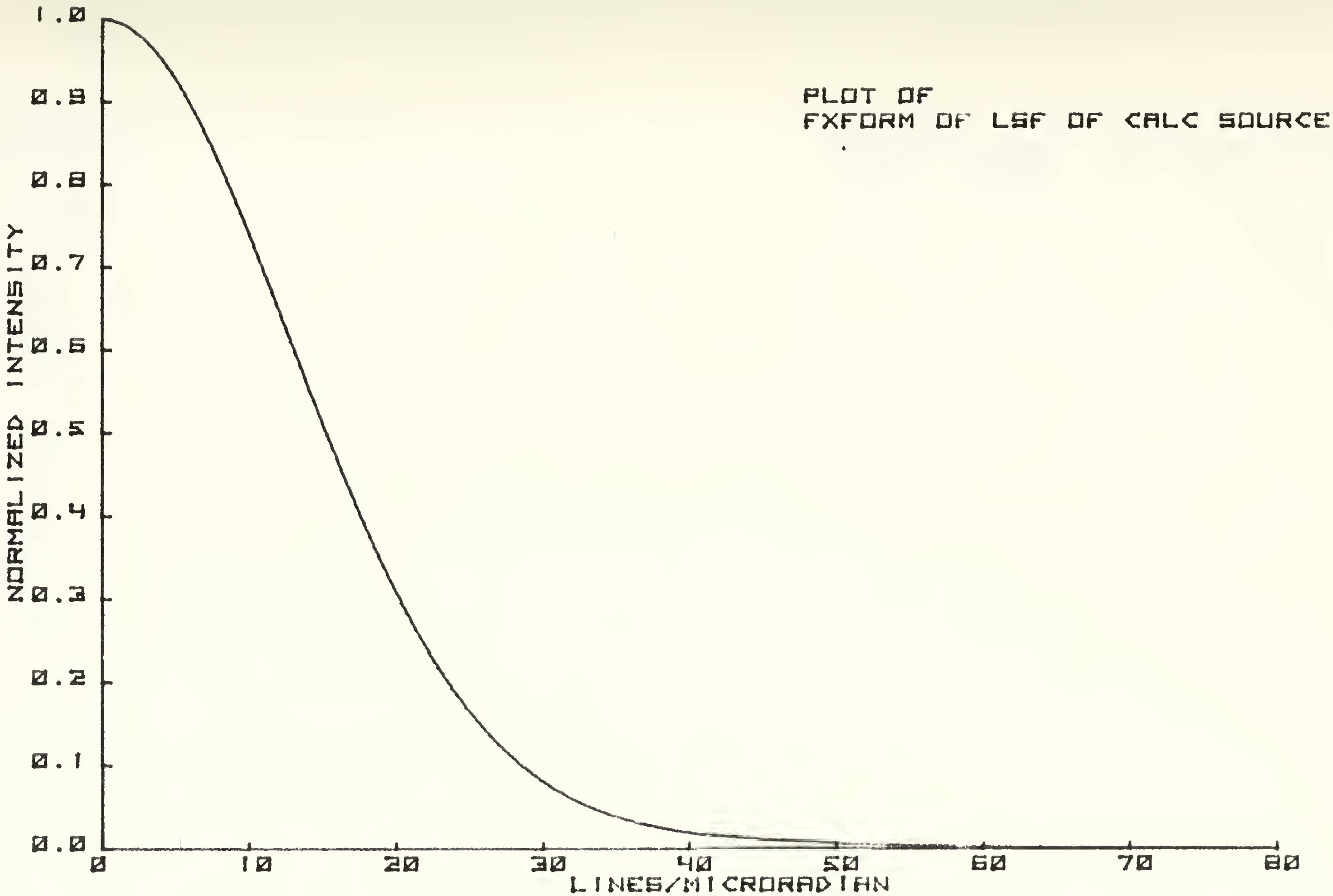

Fig. 3.22 Plot of Fourier Transform of LSF of Computed Source

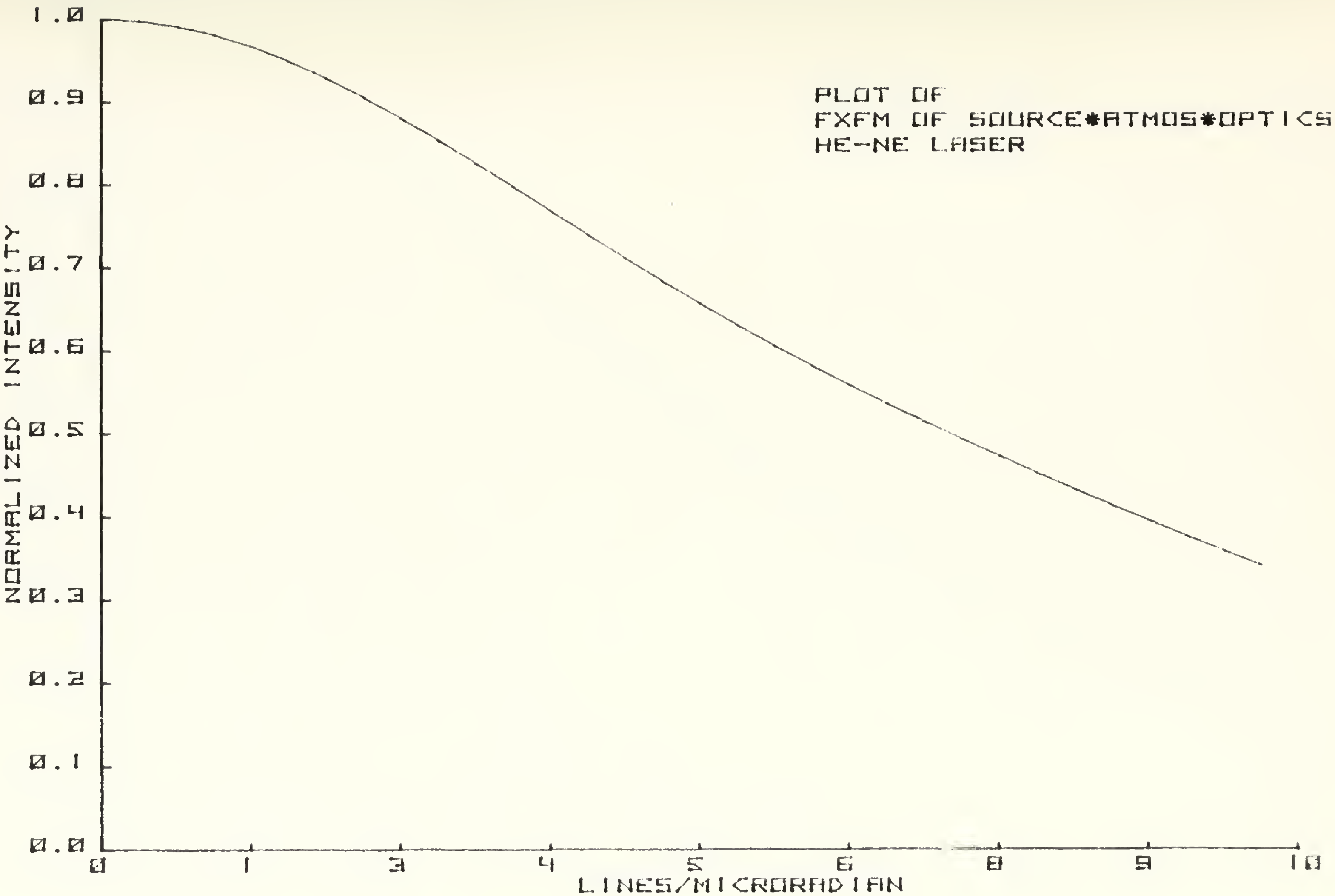

Fig. 3.23 Plot of Products of Fourier Transform of Computed Source, Optics and Atmosphere for He-Ne Laser

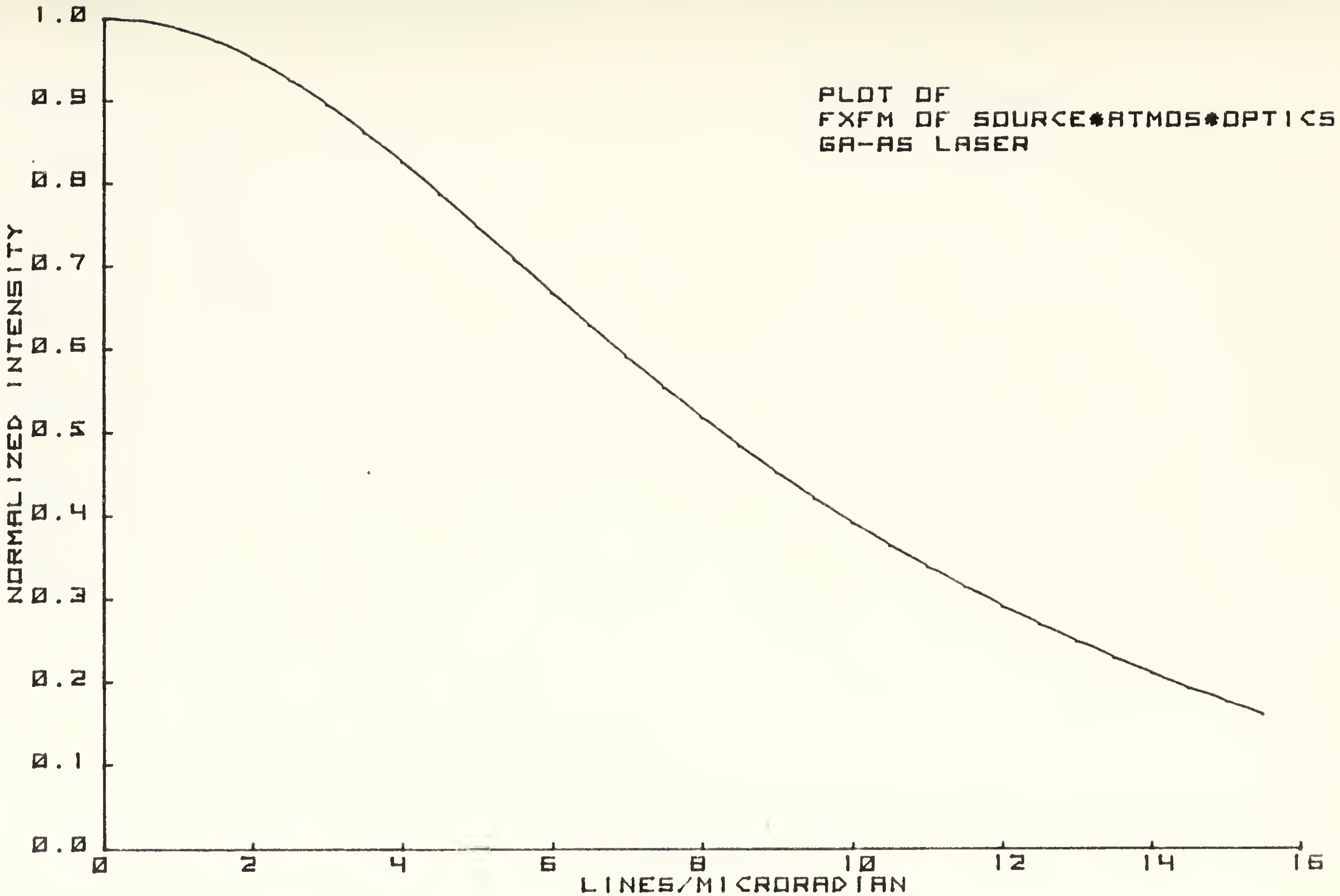

Fig. 3.24 Plot of Products of Fourier Transform of Computed Source, Optics and Atmosphere for Ga-As Laser

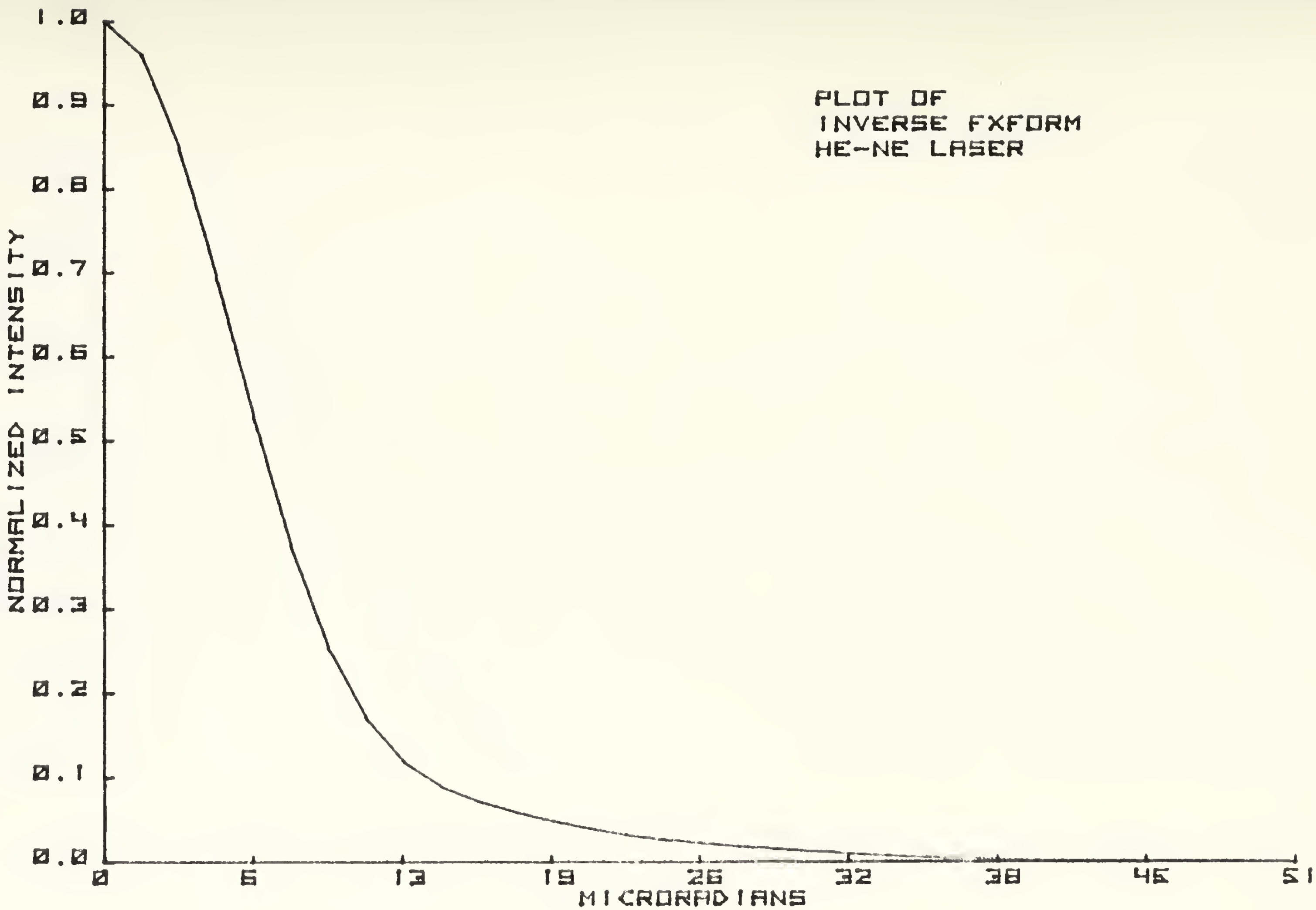

Fig. 3.25 Plot of Inverse Fourier Transform for He-Ne Laser

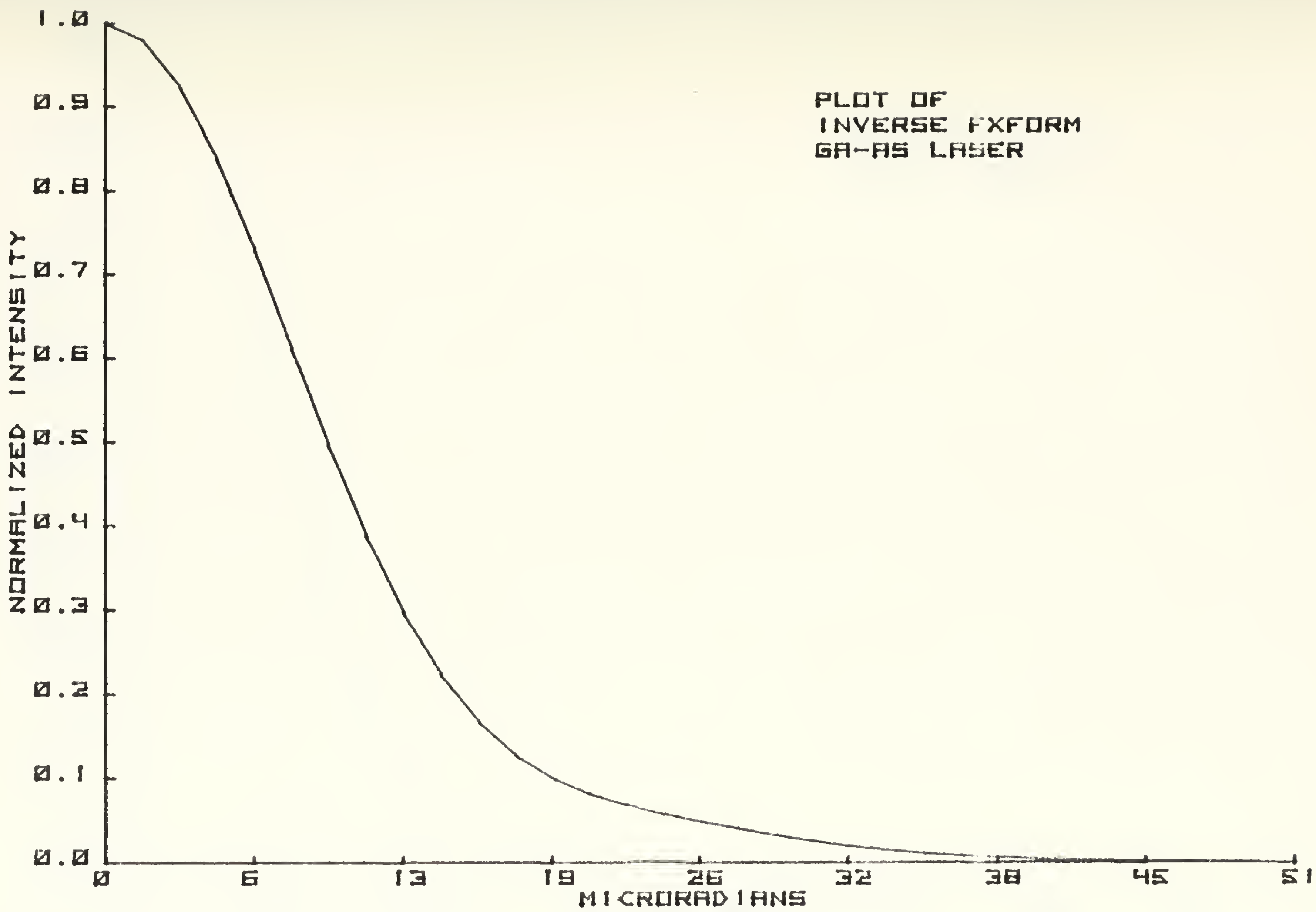

Fig. 3.26 Plot of Inverse Fourier Transform for Ga-As Laser

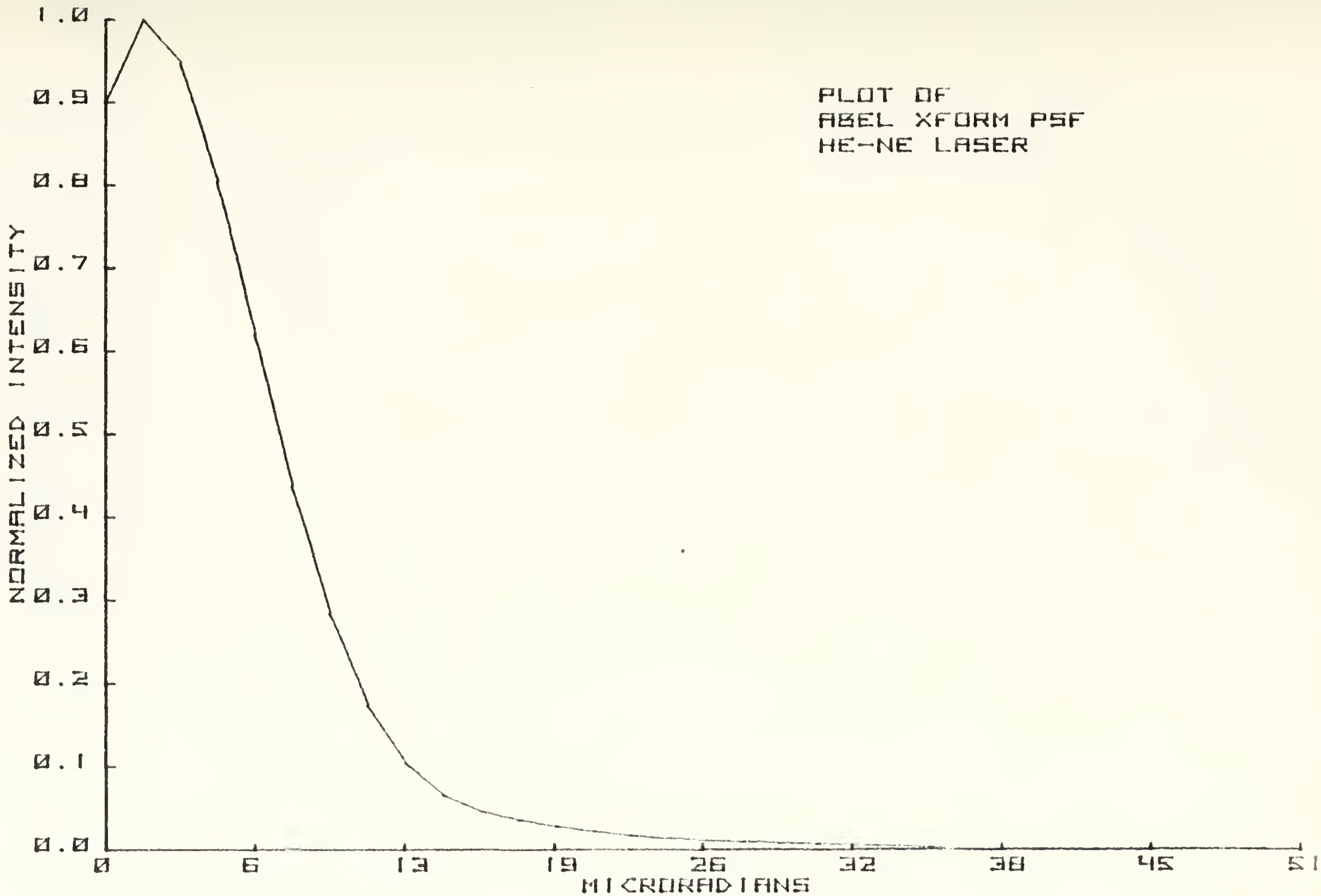

Fig. 3.27 Plot of Abel Transform for He-Ne Laser

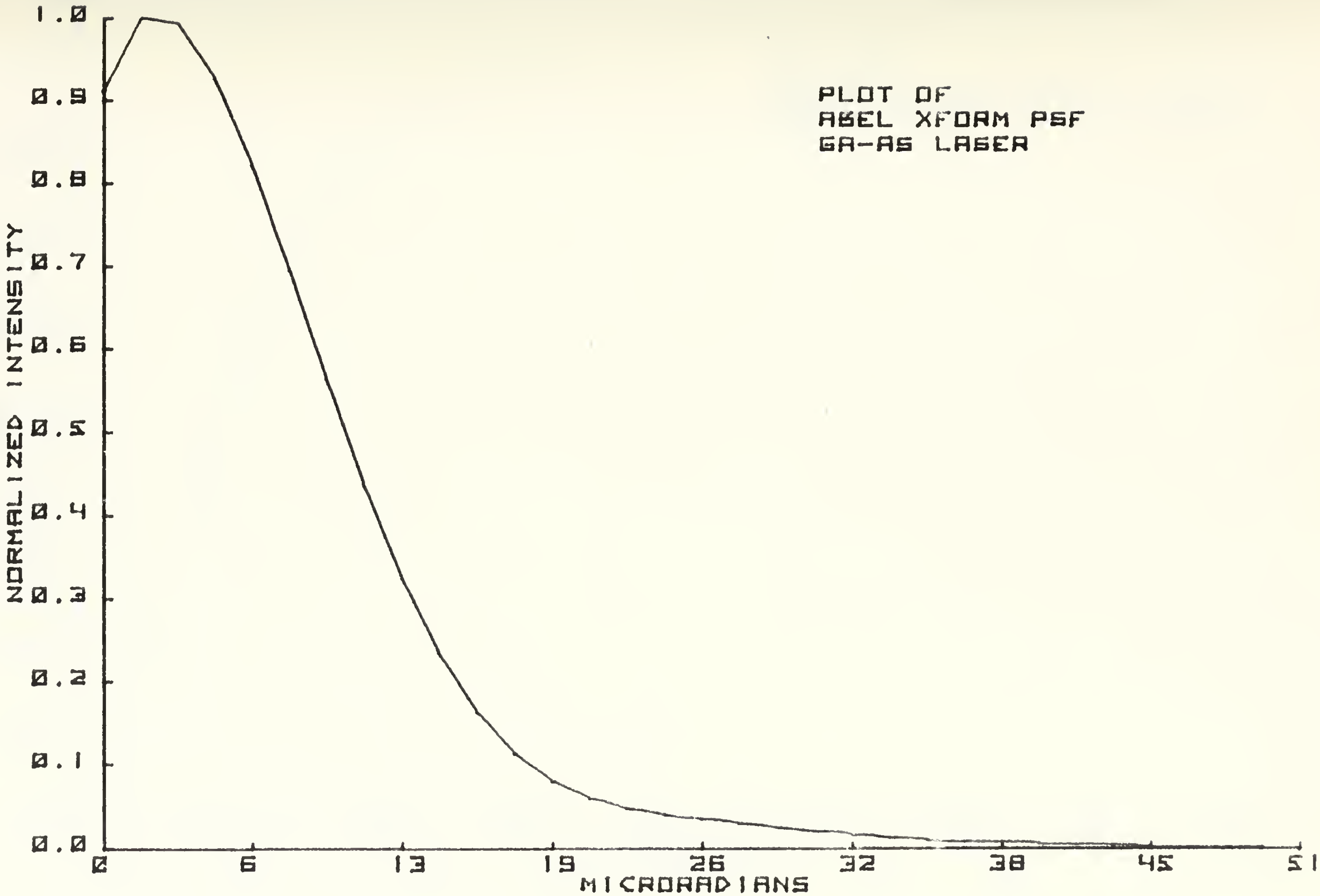

Fig. 3.28 Plot of Abel Transform for Ga-As Laser

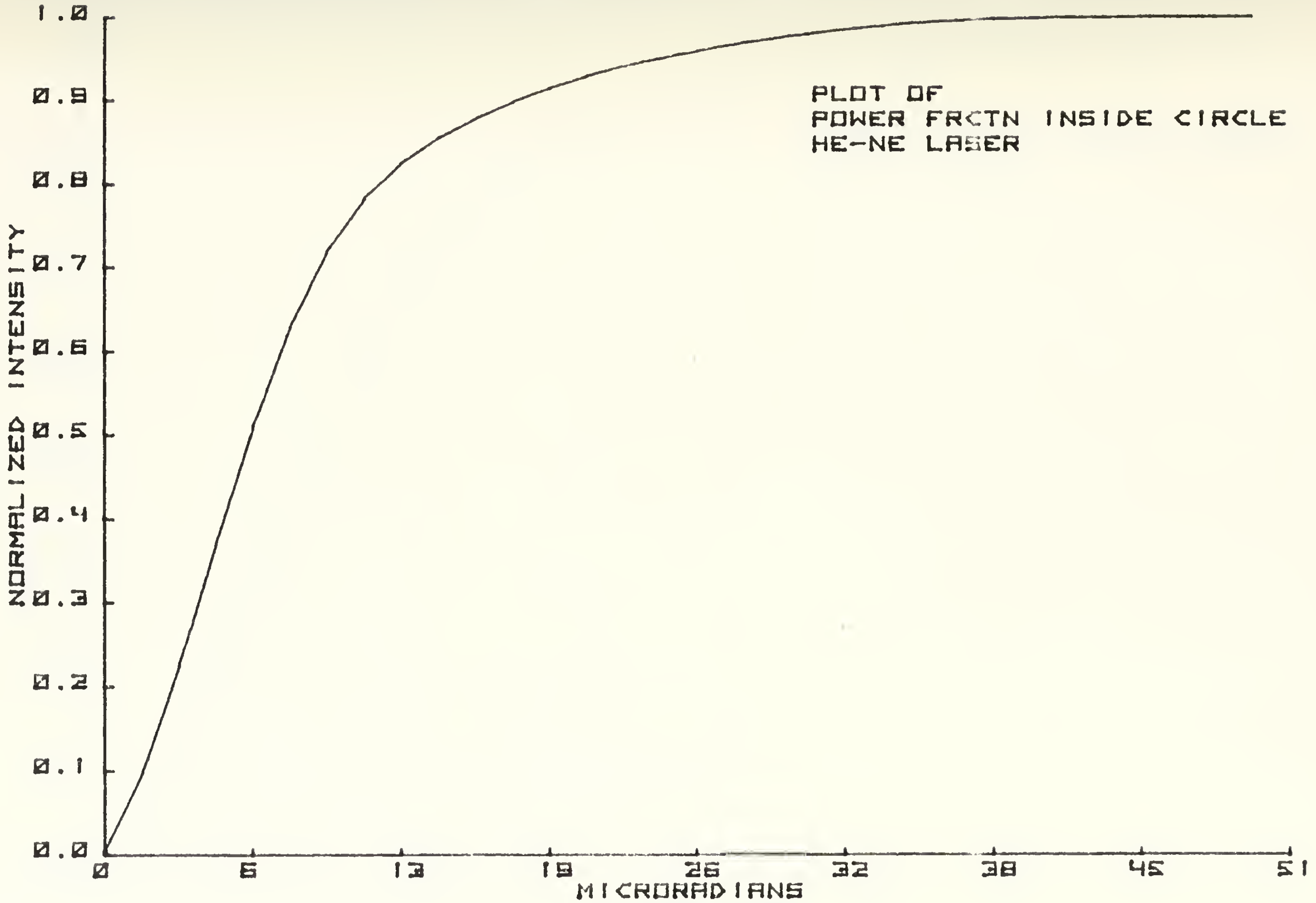

Fig. 3.29 Plot of Power Fraction Inside Circle of Radius R for He-Ne Laser

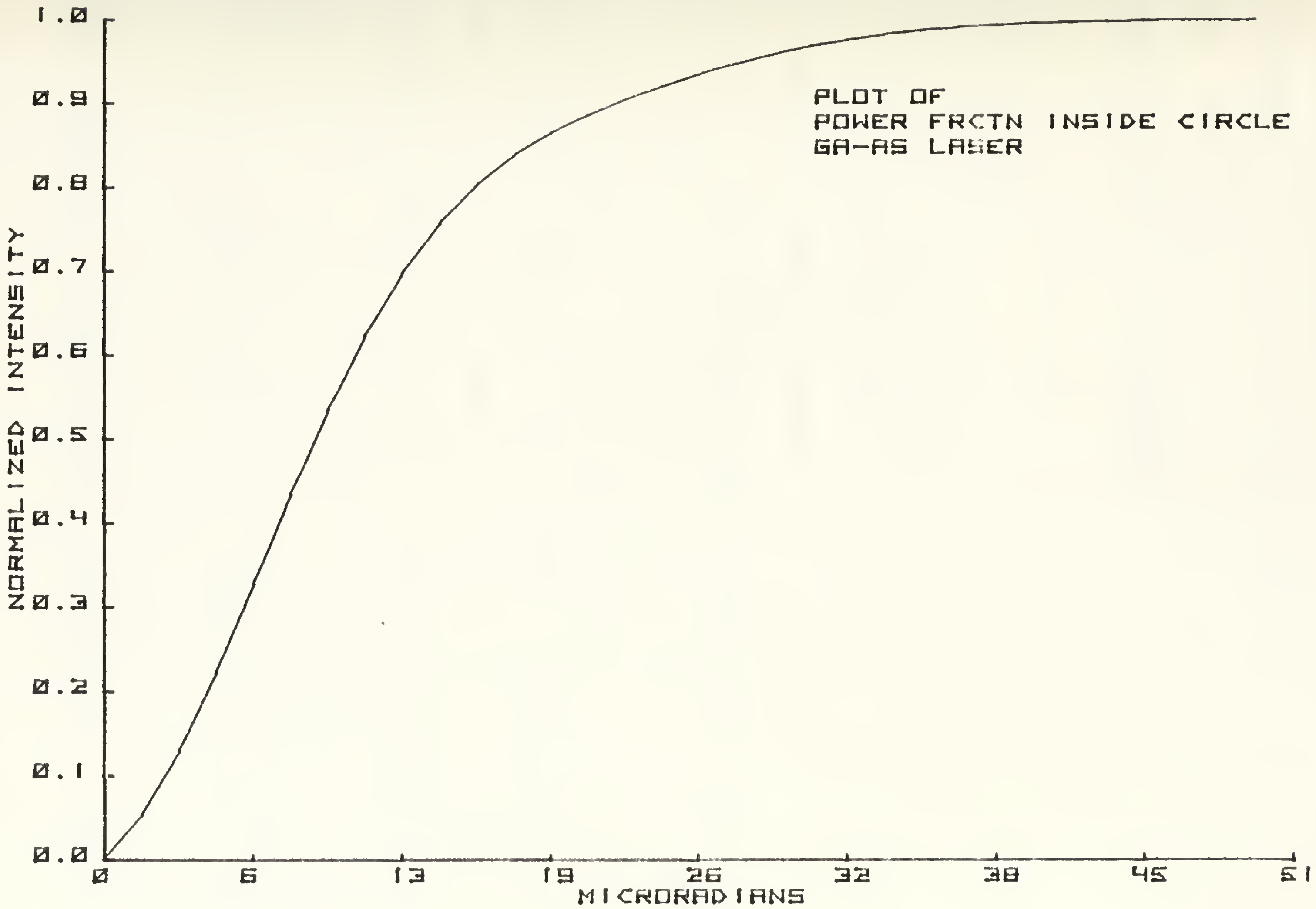

Fig. 3.30 Plot of Power Fraction Inside Circle of Radius R for Ga-As Laser

IV. COMPUTER PROGRAM IN ALGORITHMIC LANGUAGE

The program presented here is written in the algorithmic language as outlined by Graham [Ref. 9]. It is intended that the language in use here be machine independent. Comments appear in parentheses.

```
Algorithm_Laser
 (Input which laser is used.)
    Input Laser_Wavelength
    ESTRING<== Type_of_Laser
```

(Input if the particular laser source data are recorded. The following lines determine if the laser source data are already recorded. If the source data are not recorded then the program goes to the transfer subroutine to allow interfacing with the 468 to effect a data transfer. The parameters passed to the Plot subroutine have the following significance: the first parameter determines the horizontal index, for example 512 points or the number of microradians. The second parameter tells the computer which file the data are recorded on so the data can be loaded into memory. The final parameter determines the labeling of the horizontal

axis either in data points, microradians, or lines per microradian.)

```
    If Laser_Source_not_recorded then
        Call Transfer(Source_File)
        Call Plot(512,Source_File,0)
    End If
```

(Input if the particular laser scale data are recorded. Scale data are recorded with the diffraction grating in place. If the scale data need to be recorded, the subroutine Transfer is called and the scale data are stored on a separate file.)

```
    If Laser_Scale_not_recorded then
        Call Transfer(Scale_File)
        Call Plot(512,Scale_file,0)
    End If
    Input Scale_Factor
```

(Calculation of the scale factor is detailed in Chapter III. Comparison of the time base of the recorded data is necessary to ensure that the data recorded have the same time scale on the oscilloscope. If the time scales are different the program stops and displays an error message. A file is loaded, the waveform preamble is searched for the

"XUNITS" or "XINCR" oscilloscope settings and the particular data are stored. This is done for each of the 3 recorded files: source, scale, and collimated data. Collimated data refer to the same laser recorded under controlled conditions. The laser beam is projected in a collimator and recorded on tape. This data could be used to remove any non-linearities of the video recorder, electronic equipment, or receiving optics. Data elements are compared to ensure consistency of inputs.)

```
Input Laser_source
B<== pos(Laser_Source,'R:')
R1<== val('B+2',B+4')
B<== pos(Laser_Source,'IT:)
ASTRING<== Laser_Source('B+3','B+4')
Input(Laser_Scale)
B<== pos(Laser_Scale,'R:')
R2<== val('B+2',B+4')
B<== pos(Laser_Scale,'IT:')
BSTRING<== Laser_Scale('B+3','B+4')
Input(Collimated_Source)
B<== pos(Collimated_Source,'R:')
R3<== val('B+2','B+4)
B<== pos(Collimated_Source,'IT:')
```

```
   CSTRING<== Collimated_Source('B+3','B+4')
   If R1=R2=R3  then
        If ASTRING = BSTRING = CSTRING then
             Input RDATA(File_Number)
        End If
   Else
        Output 'ERROR IN TIME SCALE'
        Stop
   End If
   A<==B<==C<==0
(Array is initialized to 0)
   Do for I = 1 to 512
        YDATA(I)<== RDATA(I)
   End Do
```

(This section takes the data from the waveform message as transferred from the 468 and removes any DC background from an image as well as subtracting the minimum value in order to "zero" the data. This makes y=0 as the starting point for both calculation and plotting purposes. Data up to 51 and greater than 463 are summed and stored in B. This represents the "wings" of the curve. This process removes the DC background from an image.)

```
   Do for I = 1 to 512
```

```
        A<== A+YDATA(I)
        If I < 51 then
            B<== YDATA(I) + B
        End If
        If I > 463 then
            B<== YDATA(I) + B
        End If
    End Do
(Output for debugging)
    Output A,B
(Average of the sum of data values in the "wings".)
    D<== B/100
(This is the total sum of all data less the "wings" value.)
    A<== A-512*D
(All data are summed and stored in C. Each data
is compared with the previous total divided by 2.
This will find the index value, I, for which
the maximum value exists.)
    Do for I = 1 to 512
        YDATA(I)<== YDATA(I)-D
        C<== YDATA(I) + C
        If C < A/2 then
            J<== I
```

```
        End If
    End Do
  (Output for debugging.)
    Output C,J
 (The curve is shifted so that the maximum value starts
 at the origin of the horizontal axis. This is called
 the point spread function and is recorded as RDATA.)
    Do for K = 1 to 256
        IDATA(K)<==
YDATA((J-K+1)mod512+1)+YDATA((K+J-1)mod512+1)
    End Do
    Do for I = 257 to 512
        RDATA(I)<== 0
    End Do
    Do for I = 1 to 256
        RDATA(I)<== IDATA(I)
    End Do
 (Determines if tabular or plotted data are desired.)
    If PSF_OUTPUT_wanted then
        Call Table(RDATA)
    End If
    If PSF_PLOT_wanted then
        Call Plot(32,RDATA,1)
```

```
   End If
(Calculates the line spread function from the point
spread function.)
   Call LSF
(Determines if tabular or plotted data are desired.)
   If LSF_OUTPUT_wanted then
        Call Table(RDATA)
   Else If LSF_PLOT_wanted then
        Call Plot(32,RDATA,1)
   End If
(Calculation of Fourier Transform of LSF.)
   Input RDATA
(The maximum value of the array IDATA is at I=1. This
is put back into array RDATA and folded over so that
IDATA(257) to IDATA(512) is the mirror image of
IDATA(1) to IDATA(256). Then this array is recorded
as RDATA and the Fourier transform calculated.)
   Do for I = 257 to 512
        RDATA(I)<== 0
   End Do
   Do for I = 1 to 256
        RDATA(514-I)<== RDATA(I)
   End Do
```

```
   RDATA(257)<== RDATA(256)
   Call FXFORM
(Determines if tabular or plotted data are desired.)
   If FXFORM_OF_LSF_OUTPUT_wanted then
        Call Table(RDATA)
   Else If FXFORM_OF_LSF_PLOT_wanted then
        Call Plot(32,RDATA,2)
   End If
  (Dia of Obscur/Dia of Obj Lens in meters,RATIO=.064/.164)
  (This is the measurement of the ratio of
  diameters of obscuration to telescope.)
  (R1= Scale of data in microradians=1.60)
  (Diameter of telescope Obj Lens in meters,OBJECT=
  0.164 meters)
   If Laser_Source = He-Ne then
        Wavelength<== 6.328*e-7
   Else
        Wavelength<== 9.05*e-7
   End If
(This begins the calculation of the Fourier transform
of the optics using the Airy function. All of the
data regarding the optics are recorded on a separate file.)
(These next four statements fix the constants in the
```

argument of the optics function.)

```
   R1<== Scale_Factor
   D<== RATIO**2
   H<== 1-D
   Z<== PI*Scale_Factor*Object*1*e-6/Wavelength
   Do for I = 1 to 256
        Y<== Z*(I-1)
        If Y > 30 then
             RDATA(I) <== 0
        Else
             RDATA(I) <== (AIRY(Y)-D*AIRY(Y*RATIO))**2/H**2
        End If
        K<== I
   End Do
  (Record RDATA on separate file)
(Determines if a plot of the optics function is desired.)
   If OPTICS_FUNCTION_PLOT wanted then
        Call Plot(32,RDATA,1)
   End If
(Calculates the line spread function of the optics
function.)
   Call LSF
(Determines if a plot of the LSF of the optics function is
```

```
desired.)
   If PLOT_OF_LSF_OPTICS wanted then
        Call Plot(32, RDATA,1)
   End If
   Do for I = 257 to 512
        RDATA(I)<== 0
   End Do
(Data are folded over again and the Fourier transform
calculated.)
   Do for I = 1 to 256
        RDATA(514-I)<== RDATA(I)
   End Do
   RDATA(257)<== RDATA(256)
  (Record RDATA)
  (Fourier transform of the LSF of Optics is calculated.)
   Call FXFORM
  (Determines if plot of Fourier transform of Optics
  function is desired.)
   If PLOT_OF_FXFORM_OF_OPTICS wanted then
        Call Plot(256,RDATA,2)
   End If
  (Calculation of the quotient of two Fourier transforms.)
  (This section takes the Fourier transform of the system
```

and divides it, point by point, by the Fourier transform of the optics. This yields the modulation transfer function (MTF) of the atmosphere.)

```
Input RDATA_OF_FXFORM_OF_SOURCE
Do for I = 257 to 512
     RDATA(I)<== 0
End Do
Do for I = 1 to 256
     RDATA(514-I)<== RDATA(I)
End Do
Input RDATA_OF_FXFORM_OF_OPTICS
Do for I = 1 to 256
     RDATA(I)<== IDATA(I)/RDATA(I)
End Do
```

(Determines if a plot of the MTF is desired.)

```
If PLOT_OF_MTF_OF_ATMOS_wanted then
     Call Plot(32,RDATA,1)
End If
```

(Calculation of C_n^2 by curve fitting. This procedure is described in Chapter III.)

(All "R" variables used in this calculation are initialized to 0.)

```
R17<==R18<==R19<==R20<==R21<==R22<==R23<==R24<==R25<==0
```

(Analysis of data shows that computer time is wasted beyond 96 points.)

```
   Do for I = 1 to 96
(R18 - when summed is the (f(i))^10/3. This is the
total of the angular spatial frequencies squared.)
        R18<== ((I-1)*Scale_Factor)**(10/3)
        R17<== R17 + R18
(R19 - when summed is the (f(i))^5/3. This is
the total of the angular spatial frequencies.)
        R19<== ((I-1)*Scale_Factor)**(5/3)
        R20<== R19 + R20
(R21 - is the total number of points.)
        R21<== I
        Do While RDATA(I) > 0
(R22 - when summed is the total of the product of the
natural logarithm of each point with R19.)
             R22<== ln(RDATA(I))*R19
             R23<== R22 + R23
(R24 - when summed is the total of the natural logarithm
of each point.)
             R24<== ln(RDATA(I))
             R25<== R24 + R25
        End Do
```

```
End Do
(Output matrix values for debugging.)
  Output R17,R20,R21,R23,R25
 (Assign value to matrix.
 This section takes the above calculated values and
 sets up a matrix. The matrix equation solved is
 B = A-1*C.)
 (Create 3 matrices A(2,2),B(2,1),C(2,1).)
  A(1,1)<== R20
  A(1,2)<== R21
  A(2,1)<== R17
  A(2,2)<== R20
  INVERT_MATRIX_A
  C(1,1)<== R25
  C(2,1)<==R23
  B<== A*C
 (Output matrix values for debugging.)
  Output B(1,1),B(1,2)
 (Calculation of Cn2 using equation(2.3).)
  R22<== B(1,1)/(-21.49*Range*R20*(Wavelength**(-.33333)))
  Output 'CNSQ=',R22
 (The program now starts the prediction phase.  It
 calculates a source assumed to have a Gaussian
```

distribution by means of the computed source pattern below. Using similar Fourier transform techniques the program uses the calculated value of C_n^2 and predicts the power incident on the target.)

(Computed Source Pattern:A=A0*exp(-x**2/2*Sigma**2)

(This is an "arbitrary" Gaussian source pattern with a standard deviation for Sigma set equal to 2.)

```
 A0<== A_ZERO
 C<== Sigma*Scale_Factor
 Do for I = 1 to 256
      F<== (I-1)*Scale_Factor
      G<== F**2/(2*C**2)
      Do While G > 13
           RDATA(I) <== 0
      End Do
      RDATA(I)<==A0*exp(-G)
 End Do
```

(Record RDATA. This records the computed source.)

(Determines if plot of computed source is desired.)

```
 If PLOT_OF_COMPUTED_SOURCE_wanted then
      Call Plot(32,RDATA,0)
 End If
```

(Calculates the line spread function of the computed

source.)

```
   Call LSF
```

(Determines if a plot of the LSF of the computed source is desired.)

```
   If PLOT_OF_LSF_OF_COMP_SOURCE = 1 then
        Call Plot(32,RDATA,0)
   End If
```

(Calculation of the Fourier transform of LSF of computed source.)

```
   Do for I = 257 to 512
        RDATA(I)<== 0
   End Do
   Do for I = 1 to 256
        RDATA(514-I)<== RDATA(I)
   End Do
   RDATA(257<== RDATA(256)
   Call FXFORM
```

(Determines if a plot of the Fourier transform of the LSF of the computed source is desired.)

```
   If PLOT_OF_FXFORM_OF_COMPUTED_SOURCE_wanted then
        Call Plot(256,RDATA,2)
   End If
```

(Calculation of the product of two Fourier transforms.)

```
(This is the Fourier transform of the computed source
with the MTF of the atmosphere.)
 Do for I = 1 to 256
      RDATA(I)<==
      RDATA(I)_COMPUTED_SOURCE*RDATA(I)_FXFM_OF_MTF
 End Do
(Product of FXFORMS of Source*Atmosphere*Optics.)
 RDATA(I)<==
 RDATA(1)_PRODUCT_OF_2_FXFMS*RDATA(1)_FXFM_OF_OPTICS
 Do for I =  2 to 256
      RDATA(I)<==
      RDATA(I)_PRODUCT_OF_2_FXFMS*RDATA(I)_FXFM_OF_OPTICS
      RDATA(514-I)<==RDATA(I)
 End Do
 RDATA(257)<== RDATA(256)
(The above data predicts what the Fourier transform
of the entire system is, using the calculated
value of Cn² for the atmospheric turbulence.)
(Determines if a plot of the result of Fourier transform
is desired.)
 If PLOT_OF_FXFORM_PRODUCTS_wanted then
      Call Plot(32,RDATA,2)
 End If
```

```
(Inverse Fourier transform gives target LSF.)
 Call INVERSE_FXFORM
(Determines if a plot of the inverse Fourier transform
is desired.)
 If PLOT_OF_INVERSE_FXFROM_wanted then
      Call Plot(32,RDATA,0)
 End If
(Converts target's one dimensional LSF to a
two dimensional PSF by Abel transform.)
 Call ABEL
(Determines if a plot of the Abel transform is desired.)
 If PLOT_OF_ABEL_XFORM_wanted then
      Call Plot(32,RDATA,0)
 End If
(Calculates the fraction of power inside circle of radius
R. This predicts the fraction of power that will be
incident on target.)
 RDATA(1)<== 0.25*PI*RDATA(1)
 Do for I = 2 to 256
      RDATA(I)<== 2*PI*RDATA(I)+RDATA(I-1)
 End Do
(Determines if a plot of the fraction of power is desired.)
 If PLOT_OF_POWER_wanted then
```

```
        Call Plot(32,RDATA,0)
    End If
End Laser
```

(The following subroutines are used in the Algorithm Laser.)

Subroutine_Transfer(File_Number)

(The transfer subroutine gets raw data, both preamble and data from the waveform message sent by the 468. It processes the message by finding the minimum value of the array then subtracting this value from each element in the array. This "zeros" the array.)

```
    Output 'Ensure Equipment set up properly'
    Output 'Continue when ready'
    E<== (-1)
    Do While E = -1
```

("DATA" is the I/O buffer where the data from the 468 are sent. The status of the buffer is read while data are being transferred. When the transfer is complete, the interface is cleared.)

```
        Data<== Transferred_Data
        If Data_Transfer_Complete then
            E<== 0
        End If
```

```
End Do
Clear_Interface
N<== 1
(This is where header format stops and data begin.)
A<== pos(Data,'%')
Do for I = A+2 to 687 by 16
     Do for J = 1 to 16
          If J+I>688 then
               RJ<== 0
               N<== N+1
          End If
          RJ<== num(Data(I+J))
(The numerical values of each element are stored in YDATA.)
          YDATA(N) <== RJ
          N<== N+1
      End Do
End Do
E<== minimum(YDATA)
Do for I = 1 to 512
  YDATA(I) <== YDATA(I) -E
  RDATA(I) <== YDATA(I)
End Do
Store_RDATA(File_Number)
```

```
   Return
End Transfer
(This subroutine calculates the Line Spread Function
for the previously recorded point spread function
by using equation (1.1).)
   Subroutine_LSF
(Depending on which file number has been passed down
from the calling subroutine, either scale or source
data are loaded.)
   If File_Number = Scale_Data then
        Input Scale_Data
        RDATA<==SCALE_DATA
   Else
        Input RDATA
   End If
   Do for I = 1 to 512
        IDATA(I)<== 0
   End Do
   Do for I = 1 to 256
        IDATA(I)<== RDATA(I)
   End Do
(Plots have shown that computer time is lost and no
valuable information is gained  beyond about 24 points.)
```

```
  Do for I = 1 to 24
       J<== 1
       Output I
       Q<== IDATA(I)
       Do While R < 24
            R<== SQRT(I*I+J*J)
            Q<==
            Q+2*((1-fraction(R))*I(int(R))+fraction(R)*I(int(R)+1))
            J<== J+1
       End Do
       IDATA(I)<== Q
  End Do
  Do for I = 1 to 24
       RDATA(I)<== IDATA(I)
  End Do
  If File_Number = Scale_Data then
       Store_Scale_Data
  Else
       Store_RDATA
  End If
  Return
End LSF
```

(This subroutine calculates the Fourier transform of the given data using the Cooley-Tukey Algorithm. If the inverse statement is true, then the inverse Fourier transform is calculated.)

```
Subroutine_FXFORM
   Set_Radian_Mode
  (2**9=512 which is the number of points.)
   N<== 9
   Do for I = 1 to 512
        IDATA(I)<== 0
   End Do
   If File_Number = Scale_Data then
        Input Scale_Data
   Else
        Input RDATA
   End If
   T<== PI/2**(N-1)
        Do for J = 0 to 2**(M-1)-1
(BI is a bit inversion subroutine.)
             Call BI(J,P,N-1)
             C<== cos(P*T)
             If INVERSE_FXFORM = True then
                  Flg7<== 1
```

```
        Else
             Flg7<== 0
        End If
        P<== sin(P*T)*(1-2*Flg7)
        Do for I = 2*R0*J+1 to 2*R0*J+R0
             R1<== RDATA(I)
             R2<== RDATA(I+R0)
             R3<== IDATA(I)
             R4<== IDATA(I+R0)
             RDATA(I)<== R1+R2*C+R4*P
             IDATA(I)<== R3+R4*C-R2*P
             RDATA(I+R0)<== R1-R2*C-R4*P
             IDATA(I+R0)<== R3-R4*C+R2*P
        End Do
   End Do
   Output M
End Do
(This section is for re-ordering the block.)
Do for I = 0 to 2**N-1
   Call BI(I,J,N)
   Do While I-J < or = 0
        Do While I # J
             P<== RDATA(I+1)/SQRT(2**N)
```

```
                Z<== IDATA(I+1)/SQRT(2**N)
                RDATA(I+1)<== RDATA(J+1)
                IDATA(J+1)<== IDATA(J+1)
                RDATA(J+1)<== P
                IDATA(J+1)<== Z
            End Do
            RDATA(I+1)<== RDATA(I+1)/SQRT(2**N)
            IDATA(I+1)<== IDATA(I+1)/SQRT(2**N)
        End Do
        If File_Number = Scale_Data then
            Store_Scale_Data(File_Number)
        Else
            Store_RDATA(File_Number)
        End If
    End Do
    Set_Degrees_Mode
    Return
End FXFORM
```

(This subroutine takes the binary number P1 containing P3 bits and inverts it end for end, e.g. 010111 becomes 111010.)

```
Subroutine BI(P1,P2,P3)
    P2<== 0
```

```
    P4<== P1
    Do for Z = 1 to P3
        P4<== P4/2
        P2<== 2*P2
        If fraction(P4) # 0 then
            P2<== P2 + 1
        End If
        P4<== int(P4)
    End Do
    Return
End BI
```

(This function calculates Airy(x)=2*J1(x)/x where J1(x) is the Bessel function of order one.)

```
    Function AIRY(P1)
    If P1 < 0 then
        Output 'ERROR-ARGUMENT LESS THAN 0'
        Stop
    End If
    If P1 = 0 then
        R4<== 1
        Return R4
    End If
    R5<== 0
```

```
If P1 > 15 then
     R6<== 90 + P1/2
     R12<== 1.4*P1 + 60/P1
End If
If P1 < 5 then
     R6<== 20 + 10*P1-P1**2/3
     R12<== 6 + P1
Else
     R6<== 20 + 10*P1-P1**2/3
     R12<== 1.4*P1 + 50/P1
End If
R12<== maximum(int(R12),int(3+P1/4))
Do for M = R12 to R6 by 3
     R8<== 1*e-28
     R13<==R14<== 0
     If M/2 = int(M/2) then
          Flg10<== 0
     Else
          Flg10<== 1
     End If
     Do for J = 1 to M-2
          R15<== 2*(M-J)*R8/P1-R13
          R13<== R8
```

```
          R8<==R15
          If M-J-2 = 0 then
               R4<== R15
          End If
          If Flg10 = 0 then
               Flg10<== 1
          Else
               Flg10<== 0
          End If
          R14<== R14+2*R8*Flg10
     End Do
     R15<== 2*R8/P1-R13
     R14<== R14+R15
     R4<==R4/R14
     If (abs(R4-R5)-abs(R4*1*e-6)) < or = 0 then
          R4<== 2*R4/P1
     End If
     Return R4
     R5<== R4
  End Do
  Output 'ACCURACY NOT OBTAINED'
  Return R4
End AIRY
```

(The subroutine Abel takes a one-dimensional line spread function and calculates a two dimensional point spread function.)

```
Subroutine ABEL(File_Number)
    N<== RDATA(1)
    RDATA(1)<== 1.4*RDATA(1)-1.3*RDATA(2)+.4*RDATA(3)
    Do for I = 2 to 64
        M<== RDATA(I)
        RDATA(I)<== .4*N+.2*M-.6*RDATA(I+1)
        N<== M
    End Do
    Do for I = 1 to 64
        RDATA(I)<== RDATA(I)/(2*SQRT((I+.1)**2-I*I))
        Do for J = 1 to 64
            RDATA(I) <==
RDATA(I)+RDATA(J)/SQRT((J+.1)**2-I*I)
        End Do
        RDATA(I)<== RDATA(I)/PI
        Output I
    End Do
    Return
End ABEL
```

```
(This subroutine prints out data in tabular form.)
Subroutine_TABLE(File_Number)
   Input RDATA(File_Number)
   Do for I = 1 to 32
        Do for J = 1 to 15
        Output RDATA(16(I-1)+J)
   End Do
   Output RDATA(16*I)
   Return
End TABLE

(This subroutine plots the desired data.)
(Each time a plot is called, this subroutine
plots the particular graph and labels it.
The horizontal axis is labeled according to the first
and last parameters passed by the calling subroutine.)
Subroutine_PLOT(P1,File_Number,P3)
   Input RDATA(File_Number)
   A<== 0
   B<== P1
   C<== minimum(RDATA)
   D<== maximum(RDATA)
   XMIN<== -.1*(B-A)
```

```
XMAX<== B+.05*(B-A)

YMIN<== C-.1*(D-C)

YMAX<== D+.05*(D-C)

E<== B

F<== 10
```

(Using the value of P1, the horizontal increment for plotting and labeling is determined.)

```
If P1 = 512 then

     G<== 64

Else If P1 = 256 then

     G<== 32

Else If P1 = 64 then

     G<== 8

Else

     G<== 4

End If

Plot B,C,1
```

(This lines and places tic marks on the horizontal axis.)

```
Do for I = E to 0 by -G

     Plot I*(B-A)/E+A,C,2

     Plot I*(B-A)/E+A,C+(D-C)/150,2

     Plot I*(B-A)/E+A,C,2

End Do
```

```
(This lines and places tic marks on the vertical axis.)

  Do for I = 0 to F

       Plot A,I*(D-C)/F+C,2

  Plot A+(B-A)/150,I*(D-C)/F+C,2

  Plot A,I*(D-C)/F+C,2

  End Do

  Character_Size 1.2,1,.7,0

(This numerically labels the vertical axis.)

  Do for I = F to 0 by -1

       Plot A-.075*(B-A),I*(D-C)/F+C,1

       Label I/F

  End Do

(This sets up a character string for labeling the vertical
axis.)

  If P3 = 0 then

       ASTRING<== 'DATA POINTS"

       L<== 1

  Else If P3 = 2 then

       ASTRING<== 'LINES/MICRORADIAN'

       L<== 1/(2*L)

  End If

(This numerically labels the horizontal axis.)

  Do for I = 0 to E by G
```

```
        Plot A+(I/E-.025)*(B-A),C-.025*(D-C),1
        Label I*L
   End Do
   If P3 # 0 then
        L<== Scale_Factor
   End If
(This labels the horizontal axis.)
   Plot .4*(B-A)+A,.05*(D-C)+C,1
   Label ASTRING
   Plot -.07*(B-A)+A, .3*(D-C)+C,1
   Character_Size = 1.2,1,.7,90
   ASTRING<== 'NORMALIZED INTENSITY'
   Label ASTRING
   Character_Size = .5,1,1.5,0
(This plots data.)
   I<== 0
   Plot I,maximum(RDATA(I)),1
   Do for I = 1 to P1
        Plot I-1,RDATA(I)
   End Do
   If P3 = 0 then
        Input PLOT_LABEL
        ISTRING<== PLOT_LABEL
```

```
    End If
    Plot .6*B,.9*D,1
    Label 'PLOT OF'
    Plot .6*B,.87*D,1
    Label ISTRING
    Plot .6*B,.84*D,1
    Label ESTRING, ' Laser'
    Return
End PLOT
```

V. CONCLUSION

The work reported in this thesis supports the model predicted and measured in Crittenden, and others, for the long exposure case [Ref. 2]. The vidicon, in replacing the mechanical slit scanning system, shows no degrading of the signal data. It also demonstrates that a good approximation for the point spread function may be made by recording a single TV line through a laser spot. This line is then used to calculate the one-dimensional line spread function. The linearity of the video tape recorder is seen in the results of the MTF and C_n^2. The 468 oscilloscope is the workhorse for the entire system. It effectively displays, stores, digitizes, and transfers data on a real time basis.

The overall system does not have the capability of the system described by Crager. However, the relative simplicity of the structured programming technique coupled with a lower equipment cost demonstrates that comparable measurements and data evaluation using this equipment can be made. Further investigation should include use of the Data Precision 6000 digital waveform analyzer thereby allowing data to be sampled and analyzed on a near real time basis.

APPENDIX A

```
0: "HE-NE:SOURCE ON 9,SCALE ON 7,& COLLIMATED ON 5":
1: "GA-AS:SOURCE ON 8,SCALE ON 6,&COLLIMATED ON 4":
2: ent "SELECT CODE FOR PRINTER",A;dev "print",A
3: fmt 1,z,c;fmt 2,f8.1,z;fmt 3,f8.1
4: dim I$[1024],R[512],Y[512],C$[20],D$[20],E$[20]
5: dim I[512],A$[32],B$[20],A[2,2],B[2,1],C[2,1]
6: buf "DATA",I$,3;ina Y,I,R;0→F→W→V→U;1→L;buf "I",I$,3
7: beep;dsp "ENTER  LASER USED";wait 1500
8: ent "HE-NE=1,GA-AS=0",W
9: if W=1;"HE-NE"→E$;9→Q;jmp 2
10: "GA-AS"→E$;8→Q
11: beep;dsp E$&" SOURCE DATA RECORDED?";wait 1500
12: ent "1=YES,0=NO",V
13: if V=0 and W=1;sfg 0;cll ''TRANSFER'(Q);jmp 2
14: if V=0 and W=0;sfg 0;cll ''TRANSFER'(Q)
15: beep;ent "PLOT OF SOURCE?,1=YES,0=NO",R
16: if R=1;cll ''PLOT'(512,Q,0)
17: beep;dsp E$&" SCALE DATA RECORDED?";wait 1500
18: beep;ent "1=YES,0=NO",U
19: if U=0 and W=1;sfg 0;cll ''TRANSFER'(7→Q);jmp 2
20: if U=0 and W=0;sfg 0;cll ''TRANSFER'(6→Q)
21: beep;dsp "WHAT IS SCALE FACTOR?";wait 1000
22: ent "SCALE FACTOR=?;0 GETS PLOT",S
23: if S=0 and U#0;Q-2→Q;cll ''PLOT'(512,Q,0)
24: if S=0 and U=0;cll ''PLOT'(512,Q,0)
25: if S=0;jmp -3
26: S→L
27: if W=1;9→T→Q;jmp 2
28: 8→T→Q
29: ldf T,I$,R[*]
30: pos(I$,"R:")→B
31: val(I$[B+2,B+4])→r1
32: pos(I$,"IT:")→B
33: I$[B+3,B+4]→A$
34: ldf T-2,I$,R[*]
35: pos(I$,"R:")→B
36: val(I$[B+2,B+4])→r2
37: pos(I$,"IT:")→B
38: I$[B+3,B+4]→B$
```

```
39: ldf T-2,I$,R[*]
40: pos(I$,"R:")→B
41: val(I$[B+2,B+4])→r3
42: pos(I$,"IT:")→B
43: I$[B+3,B+4]→C$
44: if r2=r1 and A$=B$ and r2=r3 and B$=C$;jmp 2
45: beep;dsp "ERROR IN TIME SCALE";stp
46: "B":ldf Q,I$,R[*];ina Y;ara R→Y;0→A→B→C
47: for I=1 to 512
48: A+Y[I]→A;if I<51;B+Y[I]→B
49: if I>463;B+Y[I]→B
50: next I;prt A,B
51: B/100→D;A-512*D→A
52: for I=1 to 512;Y[I]-D→Y[I];C+Y[I]→C
53: if C<A/2;I→J
54: next I;prt C,J
55: for K=1 to 256
56: Y[(J-K+1)mod512+1]+Y[(K+J-1)mod512+1]→I[K]
57: next K;ina R;ara I→R
58: "IMAGE POINT SPREAD FCN"→I$;rcf 10,I$,R[*]
59: beep;ent "PSF OUTPUT=1 AND/OR CONT",Z
60: beep;ent "PSF PLOT=1 AND/OR CONT",Y
61: if Z=1;cll 'TABLE'(10)
62: if Y=1;cll 'PLOT'(32,10,1)
63: min(I[*])→r22;max(I[*])→r23
64: prt r22,r23
65: beep;ent "LSF OUTPUT=1 AND/OR CONT",Z
66: beep;ent "LSF PLOT=1 AND/OR CONT",Y
67: cll 'LSF';"IMAGE LINE SPREAD FCN"→I$;rcf 10,I$,R[*]
68: if Z=1;cll 'TABLE'(10)
69: if Y=1;cll 'PLOT'(32,10,1)
70: "CALCULATION OF FXFORM OF LSF":ldf 10,I$,R[*];ina I
71: ara R→I
72: I[1]→R[1];for I=2 to 256
73: I[I]→R[I]→R[514-I];next I
74: R[256]→R[257];rcf 10,I$,R[*]
75: cll 'FXFORM';"FXFORM OF IMAGE LSF"→I$;rcf 10,I$,R[*]
76: beep;ent "FXFORM OF LSF OUTPUT=1 AND/OR CONT",Z
77: beep;ent "FXFORM OF LSF PLOT=1 AND/OR CONT",Y
78: if Z=1;cll 'TABLE'(10)
79: if Y=1;cll 'PLOT'(32,10,2)
80: "B=DIA OF OBSCUR/DIA OF OBJ LENS IN METERS":
81: "B=0.064/.164":
82: "r1=SCALE OF DATA IN MICRORADIANS,r1→1.60":
83: "O=DIA OF OBJ LENS IN METERS;O=0.164M":
84: "W=WAVELENGTH IN METERS":
```

```
85: "CALC OF DIFFRACTION LIMIT POINT SPREAD FCN":
86: ina R;if W=1;6.328e-7→W;jmp 2
87: 9.05e-7→W
88: .064/.164→B;.164→O;L→r1
89: B^2→D;1-D→H;π*r1*O*1e-6/W→Z
90: for I=1 to 256
91: Z*(I-1)→Y;if Y>30;gto +3
92: ('AIRY'(Y)-D*'AIRY'(Y*B))^2/H^2→R[I]
93: I→K;gto +2
94: 0→R[I]
95: fxd 5;dsp R[I];next I;"OPTICS FUNCTION"→I$
96: rcf 11,I$,R[*]
97: beep;ent "OPTICS FCN PLOT=1 AND/OR CONT",Y
98: if Y=1;cll ''PLOT'(32,11,1)
99: sfg 5;cll 'LSF';"LSF OF OPTICS FCN"→I$;rcf 11,I$,R[*]
100: beep;ent "PLOT OF LSF OF OPTICS FCN=1 AND/OR CONT",Y
101: if Y=1;cll ''PLOT'(32,11,1)
102: sfg 5;ina I;ara R→I;I[1]→R[1]
103: for I=2 to 256;I[I]→R[I]→R[514-I];next I
104: R[256]→R[257];rcf 11,I$,R[*]
105: cll 'FXFORM';"FXFORM OF LSF OF OPTICS"→I$;rcf 11,I$,R[*]
106: beep;ent "PLOT OF FXFM OF OPTICS=1 AND/OR CONT",Y
107: if Y=1;cll ''PLOT'(256,11,2)
108: "CALCULATION OF QUOTIENT OF TWO FOURIER TRANSFORMS":
109: ldf 10,I$,R[*];ina I;ara R→I
110: ldf 11,I$,R[*]
111: for I=1 to 256
112: I[I]/R[I]→R[I];next I
113: "MTF OF SYSTEM"→I$;rcf 10,I$,R[*]
114: beep;ent "PLOT OF MTF OF SYS=1 AND/OR CONT",Y
115: if Y=1;cll ''PLOT'(32,10,1)
116: 0→r17→r18→r19→r20→r21→r22→r23→r24→r25
117: for I=1 to 96
118: ((I-1)*L)^(10/3)→r18
119: r18+r17→r17
120: ((I-1)*L)^(5/3)→r19
121: r19+r20→r20
122: I→r21
123: if R[I]<=0;gto "CC"
124: (ln(R[I])→r24)*r19+r23→r23
125: r24+r25→r25
126: "CC":next I
127: prt r17,r20,r21,r23,r25
128: r20→A[1,1];r21→A[1,2];r17→A[2,1];r20→A[2,2]
129: r25→C[1,1];r23→C[2,1]
130: inv A→A;mat A*C→B;flt 5;aprt B
131: "CALCULATION OF CNSQ":
132: B[1,1]/(-21.49*145*r20*W^(-.33333))→r22
133: prt "CNSQ=",r22;stp
```

```
134: cfg ;ldf 2
135: "TRANSFER":beep;dsp "ENSURE EQUIP. SET UP PROPERLY"
136: wait 1500
137: dsp "PRESS CONTINUE WHEN READY";stp
138: dsp "waiting";wait 1500
139: buf "DATA";0→Z
140: tfr 703,"DATA"
141: dsp "transferring"
142: rds("DATA")→E;if E=-1;jmp 0
143: clr 703
144: dsp "setting output"
145: pos(I$,"%")→A;1→N
146: for I=A+2 to 687 by 16
147: for J=1 to 16;if J+I>688;0→rJ;jmp 3
148: num(I$[I+J])→rJ
149: rJ→Y[N]
150: N+1→N
151: next J
152: next I
153: min(Y[*])→E;for I=1 to 512;Y[I]-E→Y[I];next I
154: ina R;ara Y→R;rcf pl,I$,R[*];ret
155: "LSF":if flg5;ldf 11,I$,R[*];jmp 2
156: ldf 10,I$,R[*]
157: ina I;ara R→I;for I=1 to 24
158: 1→J;dsp I;I[I]→Q
159: √(I*I+J*J)→R
160: 2*((1-frc(R))*I[int(R)]+frc(R)*I[int(R)+1])+Q→Q
161: J+1→J;if R<24;jmp -2
162: Q→I[I]
163: next I;ina R;ara I→R;if flg5;rcf 11,I$,R[*];cfg ;ret pl
164: ret pl
165: "FXFORM":rad;9→N;ina I;if flg5;ldf 11,I$,R[*];jmp 2
166: ldf 10,I$,R[*]
167: π/2^(N-1)→T
168: for M=1 to N;2^(N-M)→r0
169: for J=0 to 2^(M-1)-1;cll 'BI'(J,P,N-1)
170: cos(P*T)→C;sin(P*T)*(1-2*flg7)→P
171: for I=2*r0*J+1 to 2*r0*J+r0
172: R[I]→r1;R[I+r0]→r2
173: I[I]→r3;I[I+r0]→r4
174: r1+r2*C+r4*P→R[I];r3+r4*C-r2*P→I[I]
175: r1-r2*C-r4*P→R[I+r0];r3-r4*C+r2*P→I[I+r0]
176: next I;next J;dsp M;next M
177: for I=0 to 2^N-1;cll 'BI'(I,J,N)
178: if I-J>0;gto "BB"
179: if I=J;gto "INC"
180: R[I+1]/√(2^N)→P;I[I+1]/√(2^N)→Z
181: R[J+1]→R[I+1];I[J+1]→I[I+1]
182: P→R[J+1];Z→I[J+1]
```

```
183: "INC":R[I+1]/√(2^N)→R[I+1];I[I+1]/√(2^N)→I[I+1]
184: "BB":next I
185: if flg5;rcf 11,I$,R[*];deg;ret
186: deg;ret
187: "BI":0→p2;p1→p4
188: for Z=1 to p3
189: p4/2→p4;2*p2→p2
190: if frc(p4)≠0;p2+1→p2
191: int(p4)→p4
192: next Z;ret
193: "AIRY":if p1<0;beep;dsp "error-argument<0";stp
194: if p1=0;1→r4;ret r4
195: 0→r5;if p1>15;jmp 2
196: 20+10*p1-p1^2/3→r6;jmp 2
197: 90+p1/2→r6
198: if p1<5;6+p1→r12;jmp 2
199: 1.4*p1+60/p1→r12
200: max(int(r12),int(3+p1/4))→r12
201: for M=r12 to r6 by 3;1e-28→r8;0→r13→r14
202: sfg 10;if M/2=int(M/2);cfg 10
203: for J=1 to M-2;2*(M-J)*r8/p1-r13→r15;r8→r13
204: r15→r8;if M-J-2=0;r15→r4
205: cmf 10;r14+2*r8*flg10→r14;next J
206: 2*r8/p1-r13→r15
207: r14+r15→r14;r4/r14→r4
208: if abs(r4-r5)-abs(r4*1e-6)<=0;2*r4/p1→r4;ret r4
209: r4→r5;next M
210: beep;dsp "ACCURACY NOT OBTAINED";wait 1500;ret r4
211: "TABLE":ldf p1,I$,R[*]
212: for I=1 to 32;for J=1 to 15
213: wrt "print.2",R[16(I-1)+J];next J
214: wrt "print.3",R[16I];next I;ret 0→Z
215: "PLOT":ldf p2,I$,R[*]
216: 0→A;p1→B;min(R[*])→C;max(R[*])→D
217: scl A-.1(B-A),B+.05(B-A),C-.1(D-C),D+.05(D-C)
218: B→E;10→F
219: if p1=512;64→G
220: if p1=256;32→G
221: if p1=64;8→G
222: if p1=32;4→G
223: plt B,C,1
224: for I=E to 0 by -G
225: plt I(B-A)/E+A,C,2
226: plt I(B-A)/E+A,C+(D-C)/150,2
227: plt I(B-A)/E+A,C,2
228: next I
229: for I=0 to F
230: plt A,I(D-C)/F+C,2
231: plt A+(B-A)/150,I(D-C)/F+C,2
```

```
232: plt A,I(D-C)/F+C,2
233: next I;pen
234: csiz 1.2,1,.7
235: fxd 1
236: for I=F to 0 by -1
237: plt A-.075(B-A),I(D-C)/F+C,1
238: lbl I/F
239: next I
240: fxd 0
241: if p3=0;"DATA POINTS"→A$;1→L
242: if p3=2;"LINES/MICRORADIAN"→A$;1/(2*L)→L
243: for I=0 to E by G
244: plt A+(I/E-.025)(B-A),C-.025(D-C),1
245: lbl I*L
246: next I;if p3#0 and p3#2;"MICRORADIANS"→A$
247: if p3=2;S→L
248: plt .4(B-A)+A,-.05(D-C)+C;lbl A$
249: plt -.07(B-A)+A,.3(D-C)+C;csiz 1.2,1,.7,90
250: "NORMALIZED INTENSITY"→A$;lbl A$
251: csiz .5,1,1.5,0;0→I
252: plt I,max(R[*]),1
253: for I=1 to p1;plt I-1,R[I];next I
254: csiz 1.2,1,.7,0
255: if p3#0;gto +2
256: beep;ent "PLOT LABEL?",I$
257: plt .6B,.9D,1;lbl "PLOT OF"
258: plt .6B,.87D,1;lbl I$
259: plt .6B,.84D,1;lbl E$," LASER"
260: pen;cfg ;ret 0→Y
*28629
```

```
0: "COMPUTED SOURCE PATTERN":
1: ina R;1000→A;.2*L→C;trk 1;fdf 10
2: for I=1 to 256
3: L*(L-1)→F
4: F^2/(2*C^2)→G
5: if G>13;jmp 2
6: A*exp(-G)→R[I];I→K;jmp 2
7: 0→R[I]
8: next I
9: "COMPUTED SOURCE"→I$
10: rcf 10,I$,R[*]
11: beep;ent "PLOT OF COMPUTED SOURCE=1 AND/OR CONT",Y
12: if Y=1;cll 'PLOT'(32,10,0)
13: "CONVERTS PSF TO LSF":
14: cll 'LSF'
15: "LSF OF COMPUTED SOURCE"→I$
16: rcf 10,I$,R[*]
17: beep;ent "PLOT OF LSF OF COMP SOURCE=1 AND/OR CONT",Y
18: if Y=1;cll 'PLOT'(32,10,0)
19: "CALCULATION OF FXFORM OF LSF OF COMPUTED SOURCE":
20: ina I;ara R→I
21: I[1]→R[1]
22: for I=2 to 256
23: I[I]→R[I]→R[514-I];next I
24: R[256]→R[257];rcf 10,I$,R[*]
25: cll 'FXFORM'
26: "FXFORM OF LSF OF CALC SOURCE"→I$;rcf 10,I$,R[*]
27: beep;ent "PLOT OF FXFM OF CALC SRCE=1 AND/OR CONT",Y
28: if Y=1;cll 'PLOT'(256,10,2)
29: "CALCULATION OF PRODUCT OF 2 FXFORMS":
30: ldf 10,I$,R[*];ina I;ara R→I;trk 0
31: ldf 10,I$,R[*]
32: for I=1 to 256
33: R[I]*I[I]→R[I];next I
34: rcf 10,I$,R[*]
35: "TRANSFER FCN OF SOURCE*ATMOSPHERE*OPTICS":
36: ldf 11,I$,R[*];ina I;ara R→I
37: I[1]*R[1]→R[1]
38: for I=2 to 256
39: R[I]*I[I]→R[I]→R[514-I]
40: next I
41: R[256]→R[257]
42: "FXFM OF SOURCE*ATMOS*OPTICS"→I$
43: rcf 10,I$,R[*]
44: beep;ent "PLOT OF FXFM PRODUCTS=1 AND/OR CONT",Y
45: if Y=1;cll 'PLOT'(32,10,2)
```

```
46: "INVERSE FXFORM GIVES TARGET LSF":
47: sfg 7;cll 'FXFORM';cfg 7;"INVERSE FXFORM"→I$
48: rcf 10,I$,R[*]
49: beep;ent "PLOT OF INV FXFM=1 AND/OR CONT",Y
50: if Y=1;cll 'PLOT'(32,10,0)
51: "CONVERTS LSF TO PSF BY ABEL TRANSFORM":
52: cll 'ABEL';"ABEL XFORM PSF"→I$
53: rcf 10,I$,R[*];sfg 0
54: beep;ent "PLOT OF ABEL XFORM=1 AND/OR CONT",Y
55: if Y=1;cll 'PLOT'(32,10,0);stp
56: "CALCULATES FRACTION OF POWER INSIDE CIRCLE OF RADIUS R":
57: .25*π*R[1]→R[1]
58: for I=2 to 256
59: 2π*I*R[I]+R[I-1]→R[I]
60: next I
61: "POWER FRCTN INSIDE CIRCLE"→I$
62: rcf 10,I$,R[*]
63: beep;ent "PLOT OF POWER=1 AND/OR CONT",Y
64: if Y=1;cll 'PLOT'(32,10,0)
65: stp
66: "LSF":if flg5;ldf 11,I$,R[*];jmp 2
67: ldf 10,I$,R[*]
68: ina I;ara R→I;for I=1 to 24
69: 1→J;dsp I;I[I]→Q
70: √(I*I+J*J)→R
71: 2*((1-frc(R))*I[int(R)]+frc(R)*I[int(R)+1])+Q→Q
72: J+1→J;if R<24;jmp -2
73: Q→I[I]
74: next I;ina R;ara I→R;if flg5;rcf 11,I$,R[*];cfg ;ret p1
75: ret p1
76: "FXFORM":rad;9→N;ina I;if flg5;ldf 11,I$,R[*];jmp 2
77: ldf 10,I$,R[*]
78: π/2^(N-1)→T
79: for M=1 to N;2^(N-M)→r0
80: for J=0 to 2^(M-1)-1;cll 'BI'(J,P,N-1)
81: cos(P*T)→C;sin(P*T)*(1-2*flg7)→P
82: for I=2*r0*J+1 to 2*r0*J+r0
83: R[I]→r1;R[I+r0]→r2
84: I[I]→r3;I[I+r0]→r4
85: r1+r2*C+r4*P→R[I];r3+r4*C-r2*P→I[I]
86: r1-r2*C-r4*P→R[I+r0];r3-r4*C+r2*P→I[I+r0]
87: next I;next J;dsp M;next M
88: for I=0 to 2^N-1;cll 'BI'(I,J,N)
89: if I-J>0;gto "BB"
90: if I=J;gto "INC"
91: R[I+1]/√(2^N)→P;I[I+1]/√(2^N)→Z
92: R[J+1]→R[I+1];I[J+1]→I[I+1]
93: P→R[J+1];Z→I[J+1]
94: "INC":R[I+1]/√(2^N)→R[I+1];I[I+1]/√(2^N)→I[I+1]
```

```
95: "BB":next I
96: deg;ret
97: "BI":0→p2;p1→p4
98: for Z=1 to p3
99: p4/2→p4;2*p2→p2
100: if frc(p4)#0;p2+1→p2
101: int(p4)→p4
102: next Z;ret
103: "ABEL":ina R;ldf 10,I$,R[*];R[1]→N
104: 1.4*R[1]-1.8*R[2]+.4*R[3]→R[1]
105: for I=2 to 64
106: R[I]→M
107: .4*N+.2*M-.6*R[I+1]→R[I];M→N;next I
108: for I=1 to 64
109: R[I]/(2*√((I+.1)^2-I*I))→R[I]
110: for J=I+1 to 64
111: R[I]+R[J]/√((J+.1)^2-I*I)→R[I]
112: next J
113: if R[I]<.01;0→R[I]
114: R[I]/π→R[I];dsp I;next I
115: for I=65 to 512;0→R[I];next I;ret
116: "PLOT":ldf p2,I$,R[*]
117: 0→A;p1→B;min(R[*])→C;max(R[*])→D
118: scl A-.1(B-A),B+.05(B-A),C-.1(D-C),D+.05(D-C)
119: B→E;10→F;if flg0;prt A,B,C,D
120: if p1=512;64→G
121: if p1=256;32→G
122: if p1=64;8→G
123: if p1=32;4→G
124: plt B,C,1
125: for I=E to 0 by -G
126: plt I(B-A)/E+A,C,2
127: plt I(B-A)/E+A,C+(D-C)/150,2
128: plt I(B-A)/E+A,C,2
129: next I
130: for I=0 to F
131: plt A,I(D-C)/F+C,2
132: plt A+(B-A)/150,I(D-C)/F+C,2
133: plt A,I(D-C)/F+C,2
134: next I;pen
135: csiz 1.2,1,.7
136: fxd 1
137: for I=F to 0 by -1
138: plt A-.075(B-A),I(D-C)/F+C,1
139: lbl I/F
140: next I
141: fxd 0
142: if p3=2;"LINES/MICRORADIAN"→A$;1/(2*L)→L
143: for I=0 to E by G
```

```
144: plt A+(I/E-.025)(B-A),C-.025(D-C),1
145: lbl I*L
146: next I;if p3#2;"MICRORADIANS"→A$
147: if p3=2;S→L
148: plt .4(B-A)+A,-.05(D-C)+C;lbl A$
149: plt -.07(B-A)+A,.3(D-C)+C;csiz 1.2,1,.7,90
150: "NORMALIZED INTENSITY"→A$;lbl A$
151: csiz .5,1,1.5,0;0→I
152: plt I,max(R[*]),1
153: for I=1 to p1;plt I-1,R[I];next I
154: csiz 1.2,1,.7,0
155: plt .6B,.9D,1;lbl "PLOT OF"
156: plt .6B,.87D,1;lbl I$
157: plt .6B,.84D,1;lbl E$," LASER"
158: "RET":pen;cfg ;ret 0→Y
*17524
```

LIST OF REFERENCES

1. Tatarski, V.I., The Effects of the Turbulent Atmosphere on Wave Propagation, U.S. Department of Commerce, NTIS, Springfield, Va, 1971.

2. Crittenden, E.C., and others, Optical Resolution in the Turbulent Atmosphere of the Marine Boundary Layer, Naval Postgraduate School Report 61-78-003, February, 1978.

3. Crittenden, E.C., Milne, E.A., and Rodeback, G.W., Systems for Analysis for Laser Designator Spot Wander and Profile, Naval Postgraduate School Informal Report, November, 1982.

4. Crager, S.R., A Computer Program for Investigating Atmospheric Effects on Laser Designators, Master's Thesis, Naval Postgraduate School, Monterey, Cailifornia, March, 1982.

5. Lutomirski, R.F., and Yura, H.T., "Propagation of a Finite Optical Beam in an Inhomogeneous Medium", Applied Optics, v. 10, p. 1652, July, 1973.

6. Fried, D.L., "Optical Resolution through a Randomly Inhomogeneous Medium for Very Long and Very Short Exposures", Journal of Optical Society of America, v. 56, p. 1372, October, 1965.

7. Griem, H., Plasma Spectroscopy, p. 176, McGraw-Hill, 1964.

8. 468 Digital Storage Oscilloscope, v.I, Tektronix, Inc., Beaverton, Oregon, 1981.

9. Graham, N., Introduction to Computer Science, A Structured Approach, 2nd ed., West, 1982.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Commanding Officer Naval Tactical Support Activity Silver Springs, Maryland 20910	2
4.	Chairman, Department of Operations Research Code 55Fo Naval Postgraduate School Monterey, California 93940	1
5.	Professor E.A. Milne, Code 61Mn Naval Postgraduate School Monterey, California 93940	5
6.	Professor E.C. Crittenden, Code 61Ct Naval Postgraduate School Monterey, California 93940	2
7.	CDR, U.S. Army Electronics Proving Ground ATTN: STEEP-MT-T (Mr. Giorgi) Ft. Huachuca, Arizona 85613	1
8.	LCDR John H. Connor, Jr. 11 South Third Avenue Taftville, Connecticut 06380	2
9.	Chairman, Department of Physics, Code 61 Naval Postgraduate School Monterey, California 93940	1

200774

Thesis
C7158 Connor
c.1 Analysis of laser turbulence utilizing a video tape recorder and digital storage oscilloscope.